Daniel Priori

COMPARISON OF NEURAL NETWORK MODELS APPLIED TO SIZE PREDICTION OF ATMOSPHERIC PARTICLES BASED ON THEIR TWO-DIMENSIONAL LIGHT SCATTERING PATTERNS

Dissertação submetida ao Programa de Pós Graduação em Ciências da Computação para a obtenção do Grau de Mestre.
Orientador
Universidade Federal de Santa Catarina: Prof. Mauro Roisenberg Dr.
Coorientadora
Universidade de Hertfordshire: Profa. Giseli de Sousa Dra.

Florianópolis

2017

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Priori, Daniel
Comparison of neural network models applied to size prediction of atmospheric particles based on their two-dimensional light scattering patterns / Daniel Priori ; orientador, Mauro Roisenberg; coorientadora, Giseli de Sousa - SC, 2017.
100 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós Graduação em Ciência da Computação, Florianópolis, 2017.

Inclui referências.

1. Ciência da Computação. 2. Artificial Neural Networks. 3. Pattern Recognition. 4. Deep Learning Autoencoders. 5. Two-dimensional Light Scattering Patterns. I. Roisenberg, Mauro. II. Sousa, Giseli de. III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Computação. IV. Título.

Daniel Priori

COMPARISON OF NEURAL NETWORK MODELS APPLIED TO SIZE PREDICTION OF ATMOSPHERIC PARTICLES BASED ON THEIR TWO-DIMENSIONAL LIGHT SCATTERING PATTERNS

Esta dissertação foi julgada aprovada para obtenção do título de "Mestre" e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Florianópolis, 29 de março de 2017.

_____
Profª. Carina Friedrich Dorneles, Drª.
Coordenadora do Programa

_____
Profª. Giseli de Sousa, Drª.
Coorientadora
Universidade de Hertfordshire

Banca Examinadora:

_____
Prof. Mauro Roisenberg, Dr.
Universidade Federal de Santa Catarina
Orientador

_____
Profª. Yi Sun, Drª.
Universidade de Hertfordshire (Videoconferência)

_____
Profª. Silvia Modesto Nassar, Drª.
Universidade Federal de Santa Catarina

_____
Prof. Ricardo Azambuja Silveira, Dr.
Universidade Federal de Santa Catarina

This work is dedicated to my lovely daughter Luisi, my dear wife Elisandra and my dear parents

ACKNOWLEDGMENT

First of all, I would like to thank my advisor Mauro Roisenberg and my co-advisor Giseli de Sousa for their immense work in guiding and assisting me with patience and dedication in this new trail of uncertainties, challenges and at the same time, fascinating universe of computing, specially the area of computational intelligence. You were exceptional and I will be eternally grateful for all you have done. I look forward to work with you again on new challenges.

Thanks also to University of Hertfordshire group, Dr. Yi Sun, Dr. Neil Davey, Dr. Evelyn Hesse and Dr. Christopher Stopford, who provided the data used in this study as well as assisted me with mastery and dedication over of my journey. It has been an honor to work with you and I look forward to working on new discoveries.

I am also grateful to the other lectures and members of UFSC postgraduate program who supported me and helped me in the best way possible.

I thank also to my friends who supported me in some steps of this journey and to my family, speacilly my parents, my brother Lucas and their wife Samantha who helped me always when I needed.

And last but not least, I thank to my wife Elisandra, who accompanied me the whole process from the beginning, specially when I had to work hard. We jumped the obstacles. When I had doubts or during my weariness, she was there supporting me and giving me strength to continue. In the uncertainties and bifurcations, she was there with me, always advising me and trusting me. And in the achievements, she was there always to hug me and celebrate with me, filling my heart with joy and relief. My dear wife, this work is yours too.

RESUMO

A obtenção do tamanho projetado de partículas atmosféricas prismáticas é de imensa importância em diversos aspectos da vida prática. Partículas expelidas por erupções vulcânicas podem por em risco a aviação civil e militar. Cristais de gelo presentes em nuvens, dependendo de seu tamanho e formato, alteram as propriedades radiantes das nuvens que podem, por sua vez, afetar significativamente os modelos climáticos. Uma forma indireta de se obter informações sobre as partículas prismáticas é através da utilização de instrumentos que registram padrões bidimensionais de dispersão de luz. Estas imagens podem ser utilizadas para caracterizar uma partícula cristalina, fornecendo informações sobre tamanho, razão de proporção, forma, concavidade e rugosidade. Neste trabalho procurou-se aplicar técnicas de Aprendizado de Máquina, em especial alguns modelos de redes neurais artificiais e técnicas de análise de dados, de forma a encontrar um modelo que apresente um desempenho satisfatório na tarefa de predição do tamanho projetado das partículas cristalinas. Os modelos de redes neurais testados foram do tipo Feed Forward Multi-Layer Perceptron com regularização Bayesiana, as redes neurais do tipo Função de Base Radial, e as redes Deep Learning do tipo Autoencoders, a qual também foi aplicada com o propósito de redução dimensional. Também foram testadas as técnicas de análise de dados de redução dimensional utilizando Análise de Componentes Principais e invariância à rotação das imagens através da Transformada Rápida de Fourier. Os modelos apresentados foram aplicados a uma série de imagens e seus resultados comparados e analisados. O modelo desenvolvido que utiliza conceitos de Deep Learning com técnicas de Autoencoder foi aquele que obteve os melhores resultados (performance de 0.9914), em especial na predição de tamanho projetado para as partículas menores, as quais tiveram maiores dificuldades de predição nos outros modelos propostos nesse trabalho.

Palavras-chave: Redes Neurais Artificiais, Padrões de Dispersão de Luz Bidimensionais, partículas atmosféricas, reconhecimento de padrões, Transformada Rápida de Fourier, Função de Base Radial, Deep Learning, Autoencoders, Análise de Componentes Principais

RESUMO ESTENDIDO

CONTEXTUALIZAÇÃO

A obtenção do tamanho projetado de partículas atmosféricas prismáticas é de imensa importância em diversos aspectos da vida prática. Cristais de gelo presentes em nuvens, dependendo de seu tamanho e formato, alteram as propriedades radiantes das nuvens que podem, por sua vez, afetar significativamente os modelos climáticos. Uma forma indireta de se obter informações sobre as partículas prismáticas é através da utilização de instrumentos que registram padrões de imagens do tipo *Two-Dimensional Light Scattering* (2DLS). Estas imagens podem ser utilizadas para caracterizar uma partícula cristalina, fornecendo informações sobre tamanho, razão de proporção, forma, concavidade e rugosidade.

A partir das imagens obtidas dos padrões 2DLS das partículas, é desejável extrair algumas características, como proporção, forma e tamanho da partícula. No entanto, a característica que desejamos determinar neste trabalho é o tamanho de partícula projetado visto pela luz incidente a partir do padrão 2DLS. Este é um parâmetro importante porque, se desenharmos o tamanho projetado, obtemos assim uma indicação do tamanho de partícula real.

Em outras palavras, o principal problema envolvendo este trabalho é sobre a extração de características de partículas atmosféricas através de imagens de padrões bidimensionais de dispersão de luz. Portanto, é um problema de reversão de um processo físico em que as partículas atmosféricas são capturadas por instrumentos que geram padrões de imagens das quais podemos extrair características das partículas correspondentes. Assim, neste trabalho, propomos quatro modelos computacionais usando técnicas de redes neurais agregadas a outras técnicas de tratamento de dados e invariância de rotação de padrões de dispersão de luz para poder prever características de partículas não esféricas (especialmente hexagonal) a partir de um banco de dados com padrões de imagens de 162 partículas fornecidas por Stopford (2010) através do Centro de Pesquisa Atmosférica e de Instrumentação da Universidade de Hertfordshire, Inglattera.

OBJETIVOS

O objetivo deste trabalho é propor e comparar quatro modelos computacionais usando redes neurais (NN) que podem efetivamente prever o tamanho projetado de partículas atmosféricas. Para atingir o objetivo deste trabalho, os seguintes objetivos devem ser preenchidos:

- Propor um modelo computacional eficiente usando técnicas de rede neural que podem prever tamanhos de partículas projetadas através de padrões 2DLS;
- Testar e comparar os modelos propostos através de experimentos com o conjunto de dados de padrões de partículas 2DLS;

METODOLOGIA

Os quatro modelos computacionais propostos tem como entrada os padrões de imagem 2DLS e utilizam técnicas de tratamento de dados como normalização, redução dimensional e transformação do domínio espacial para domínio de frequência em cada padrão de imagem através da Transformada Rápida de Fourier (FFT). O propósito para transformação para o dominío de frequência é para evitar problemas de rotação das imagens, pois uma imagem rotacionada, mesmo sendo do mesmo padrão, para a rede neural pode ser encarada como sendo um padrão diferente.

Para redução dimensional, os modelos 1, 2 e 3 utilizam a técnica de Análise de Componentes Principais (PCA). O modelo 4 utiliza Autoencoders. Após o tratamento de dados específico, cada modelo utiliza técnicas de redes neurais diferente: o modelo 1 utiliza redes neurais *Multi-Layer Perceptron* (MLP) com 50 nós na camada escondida e regularização Bayesiana por retro propagação como função de treinamento. O modelo 2 utiliza duas redes semelhantes ao modelo 1 para regressão e uma MLP para classificação entre dois tipos de padrões de partículas. O modelo 3 utiliza Função de Base Radial com 4 ou 5 mil neurônios na camada escondida. Por último, o modelo 4, utiliza Deep Learning com dois Autoencoders e uma rede de função linear. Todos os quatro modelos tem como saída a predição de tamanho projetado das partículas.

EXPERIMENTOS E RESULTADOS

Nos experimentos utilizamos o mesmo dataset para todos os modelos propostos contendo padrões de imagem de 162 partículas at-

mosféricas. Cada partícula contém 133 orientações, onde cada orientação é um padrão de imagem 2DLS. Ou seja, ao todo o dataset é composto por 21546 padrões de imagem como a entrada de dados de todos os modelos e a saída é a predição do tamanho projetado de cada partícula do dataset de teste (70% do dataset foi utilizado para treino nas redes e 30% para teste).

Os resultados do modelo 1 em média foi satisfatório com uma taxa de performance (R^2) de 0.9853. Entretanto, para as partículas menores, a performance é visivelmente insatisfatória. Já o modelo 2, com duas redes de regressão e uma de classificação, não obteve uma performance satisfatória (NN 1: 0.9812 — NN 2: 0.2451) tendo como principal problema a classificação de padrões de partículas entre problemáticas (as quais obtiveram erro acima de 20% na predição no modelo 1) e não problemáticas. O modelo 3 consegue ter uma taxa melhor com $R^2 = 0.9891$. Por fim, os resultados mais satisfatórios foram o do modelo 4, com $R^2 = 0.9914$ e uma maior precisão de predição nas partículas menores, se comparado aos demais modelos.

Palavras-chave: Redes Neurais Artificiais, Padrões de Dispersão de Luz Bidimensionais, partículas atmosféricas, reconhecimento de padrões, Transformada Rápida de Fourier, Função de Base Radial, Deep Learning, Autoencoders, Análise de Componentes Principais

ABSTRACT

Obtaining the projected size of atmospheric prismatic particles is of immense importance in many aspects of practical life. Particles expelled by volcanic eruptions may threat to civil and military aviation. Ice crystals present in clouds, depending on their size and shape, can modify the radiant properties of clouds that can significantly affect the climate models. An indirect way of obtaining information on prismatic particles is through the use of instruments that record two-dimensional light scattering patterns. These images can be used to characterize a crystalline particle, providing information on size, aspect ratio, shape, concavity and roughness. In this work we tried to apply Machine Learning techniques, especially some models of artificial neural networks and techniques of data analysis, in order to find a model that presents a satisfactory performance in the task of predicting the projected size of the crystalline particles. The models of neural networks tested were Feed Forward Multi-Layer Perceptron neural network with Bayesian regularization, Radial Basis Function neural network and Deep Learning network with Autoencoders, which was applied for dimensional reduction purpose as well. We also tested techniques of data dimensional reduction such as Principal Component Analysis and techniques for image rotation invariance such as the Fast Fourier Transform. The presented models were applied to a series of images and their results were compared and analysed. The developed model which used concepts of Deep Learning with techniques of Autoencoder was the one that obtained the best results (0.9914 of performance), and especially in the prediction of projected size of the smaller particles, which had greater difficulties of prediction when using the other models proposed in this work.

Keywords: Artificial Neural Networks, Two-dimensional Light Scattering Patterns, atmospheric particles, pattern recognition, Fast Fourier Transform, Radial Basis Function, Deep Learning, Autoencoders, Principal Component Analysis

LIST OF FIGURES

Figure 1 SID-2 Instrument with electronics exposed. Source: Stopford (2010) 25
Figure 2 SID-2 instrument illustrated with the optical path and a 2DLS example recorded 26
Figure 3 A selection of ice analogues imaged by Scanning Electron Microscopy (SEM). Source: Stopford (2010) 26
Figure 4 Examples of hexagonal prism particle shapes with their respective 2DLS patterns in a corresponding orientation 27
Figure 5 Diagram of general processes for all models..... 38
Figure 6 Examples of diffraction patterns in disk format 40
Figure 7 Diffraction pattern images of the same particle with different rotations..... 42
Figure 8 Frequency values of the same 2DLS with different rotations 43
Figure 9 Amount of components that explain the variation in dataset. The y-axis shows the percentage of explained variation and x-axis shows the amount of components necessary to extract the correspondent variation. 45
Figure 10 Autoencoder scheme used for Dimensional Reduction .. 46
Figure 11 MLP Feedforward Neural Network scheme 47
Figure 12 Diagram of prediction process performed by the first proposed model 48
Figure 13 Model 1 plot predicted size against real pattern size.... 50
Figure 14 Model 1 plot predicted size against real pattern size of a squared small particle 51
Figure 15 2DLS patterns from different size particles with the same aspect ratio and same orientation 52
Figure 16 Histogram with the number of problematic particle orientations..... 52
Figure 17 Problematic patterns that have similar beta angles..... 53
Figure 18 Training and test plot prediction of a unique and small compact particle using Model 1 54
Figure 19 Diagram of the model 2 which uses two neural networks for prediction and one for classification 55

Figure 20 Result of training and test with a classification network and two prediction networks 57
Figure 21 Prediction process diagram with model 3 approach 59
Figure 22 Plot example of test process with model 3 60
Figure 23 Plot example of of a small particle from test process with model 3 61
Figure 24 Schematic of an Autoencoder with details 62
Figure 25 Diagram of the complete model 4 that use concepts of Deep Learning with Sparse Autoencoders 67
Figure 26 Diagram of the neural network model developed with two Autoencoders and one network with linear transfer function separately 69
Figure 27 Network model constructed from the structures of the two Autoencoders and the network with linear transfer function .. 70
Figure 28 Plot showing predicted size with projected size of test process with model 4 71
Figure 29 Plot of prediction results for the test set with model 1 . 74
Figure 30 Plot of prediction result for particle 1 used in the test set with model 1 74
Figure 31 Results from a small particle and a large particle 75
Figure 32 Runs of the network with diffraction contrast enhancement of particles images 77
Figure 33 Network results for a square small particle when the network was trained with all training dataset and tested with all testing dataset 78
Figure 34 Plot test results for the particle 112 with all its 133 orientations 78
Figure 35 Confusion matrix of model 2 supervised NN for classification between problematic and non-problematic patterns 80
Figure 36 Training and test plots of the two prediction networks of model 2 81
Figure 37 Test results using model 3 with RBF neural networks .. 82
Figure 38 Test results of a squared small particle using model 3 .. 83
Figure 39 Test results with model 4 84
Figure 40 Test results with model 4 of the smaller and squared particle 84
Figure 41 Prediction comparison of all models with all test dataset. 88

Figure 42 Prediction comparison of all models with a smaller squared particle 89
Figure 43 Prediction comparison of all models with a larger particle 90

LIST OF TABLES

Tabela 1 Summary of related works including their goals and methods applied on each of them 35
Tabela 2 Statistics from dataset and correlation means between size and image patterns 41
Tabela 3 Parameter values used in the two Autoencoders 71
Tabela 4 Performance values of each particle in test dataset for all models 85
Tabela 5 Performance and error values of each model 87
Tabela 6 Statistic of the model 4 prediction 91
Tabela 7 Computational time using model 4 (in seconds) 94

LIST OF ABBREVIATIONS

2DLS	2-Dimensional Light Scattering
AE	Autoencoder
DBN	Deep Belief Network
FFT	Fast Fourier Transform
GA	Genetic Algorithm
GRNN	General Regression Neural Network
KL	The Kullback-Leibler divergence
LDA	Linear Discriminant Analysis
LL	Linear Layer
MDLS	Multiangle Dynamic Light Scattering
MLP	Multi-layer Perceptron
MT	Main Technique
NMSE	Normalised Mean Square Error
NN	Neural Network
PC	Personal Computer
PCA	Principal Component Analysis
RBF	Radial Basis Function
RFE	Recursive Feature Elimination
RI	Rotation Invariance
RTDF	Ray Tracing with Diffraction on Facets
SEM	Scanning Electron Microscopy
SID	Small Ice Detector
UFS	Univariate Feature Selection
ZM	Zernike Moments

SUMMARY

1	**INTRODUCTION**	25
1.1	CONTEXTUALISATION	25
1.2	OBJECTIVES	28
1.3	CONTRIBUTIONS	29
1.4	ORGANIZATION OF THE THESIS	30
2	**RELATED WORKS**	31
3	**METHODOLOGY AND DEVELOPMENT**	37
3.1	DATASET	39
3.1.1	**Dataset correlations**	41
3.2	NORMALISATION	41
3.3	SPLIT AND RANDOM ROTATE	42
3.4	ROTATION INVARIANCE	42
3.5	FEATURE SELECTION AND DIMENSIONAL REDUCTION	43
3.5.1	**Dimensionality Reduction with Principal Component Analysis (PCA)**	44
3.5.2	**Dimensionality reduction with Autoencoders**	45
3.6	PROPOSED MODELS	46
3.6.1	**Model 1: MLP FF with a Hidden Layer**	47
3.6.2	**Model 2: Two Multi-Layer Feed-Forward Neural Networks applied for Regression and One for Classification problems**	54
3.6.3	**Model 3: Radial Basis Function (RBF) Network**	57
3.6.4	**Model 4: Deep Learning with Sparse Autoencoders**	62
4	**EXPERIMENTS AND RESULTS**	73
4.1	ENHANCEMENT OF CONTRAST INTENSITIES	76
4.2	TRAINING AND TEST PARTICLES INDIVIDUALLY	78
4.3	EXPERIMENTS WITH MODEL 2	79
4.4	EXPERIMENTS WITH MODEL 3	81
4.5	EXPERIMENTS WITH MODEL 4	83
4.6	COMPARISONS AND DISCUSSION	85
5	**CONCLUSIONS**	93
5.1	MAIN CONTRIBUTIONS	93
5.2	FUTURE WORKS	95
	REFERENCES	97

1 INTRODUCTION

1.1 CONTEXTUALISATION

Atmospheric clouds influence the climate through radiation (distribution and absorption of solar and thermal radiation) and other physical processes that impact the Earth's radiation balance and affect climate change. Such clouds are a source of significant uncertainty in climate models as presented in the Fifth Assessment Report of the Intergovernmental Panel on Climate Change (CLIMATE..., 2013). This applies specifically to ice or mixed-phase cloud particles (these being both ice crystals and supercooled droplets), since the radiant properties of such clouds are dependent on the relative abundance of the crystals and droplets, their size spectra and, in particular, the various shapes of the particles presented in the clouds (BARAN, 2009).

To be able to understand the radiative transfer properties of such particles, a detailed knowledge of their shapes and sizes was required. The imaging methods, for example (LAWSON et al., 1998), are widely used to obtain in situ morphological data of atmospheric particles. However, for small particles, optical anomalies and depth of field constrain the information obtained. Such restrictions do not apply to the detection of dispersion patterns. Therefore, suitable detection instruments such as the Small Ice Detector (SID) (KAYE et al., 2008) have been developed (see Figures 1 and 2).

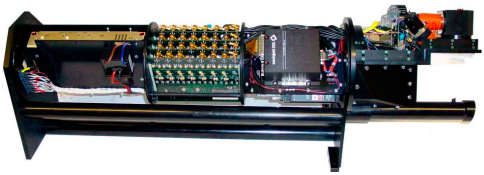


Figure 1: SID-2 Instrument with electronics exposed. Source: Stopford (2010)

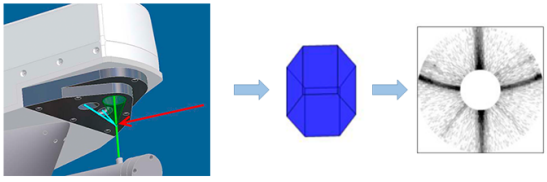


Figure 2: SID-2 instrument illustrated with the optical path and a 2DLS example recorded. Red arrow indicates the incoming particle. Light blue lines indicate the path from scattering volume to trigger detectors. Green cone indicates the captured forward scattered light. Once the SID record the 2DLS pattern from a particle, this pattern we can see like this example here, in polar coordinates. Source: Stopford (2010)

However, while conventional pattern recognition methods can be easily used for the group of images recorded by Two-Dimensional Light Scattering (2DLS) patterns, in broad-particle classes (KAYE et al., 2008), the inversion of the necessary patterns to obtain quantitative morphological data is much more complicated. Thus, the creation of datasets with known particle dispersion morphological patterns is extremely useful for particle characterization.

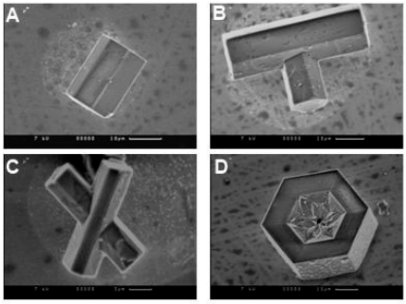


Figure 3: A selection of ice analogues imaged by Scanning Electron Microscopy (SEM). Source: Stopford (2010)

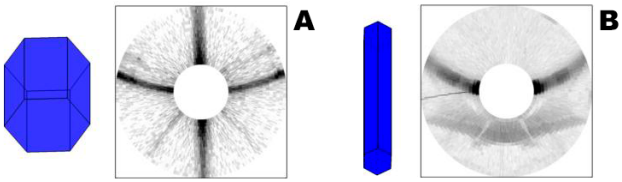


Figure 4: Examples of hexagonal prism particle shapes with their respective 2DLS patterns in a corresponding orientation and in polar coordinates. In plot A, we can see a large compact particle with 38μm size with an aspect ratio close to 1 (squared). In plot B, we can see a particle with 14μm size with an aspect ratio of 8. Source: Stopford (2010)

While some particles are known as a wide variety of shapes (see Figure 3) such as pillars, plates, rosettes, aggregates and variations (for example, hollow or rough and / or rounded crystals) (BARAN, 2009), the dataset provided by (Dr Chris Stopford of) the University of Hertfordshire's Centre for Atmospheric and Instrumentation Research is composed by intact hexagonal prisms (i.e., undistorted) which vary in size and aspect ratio (as shown the two examples of hexagonal prisms in Figure 4). Aspect ratio in this work is defined as the ratio of the particle length by its diameter, where the diameter is twice the length of the edge of a hexagonal facet. Because these particles are not spherical, it is common for the particle to be captured at different angles of rotation even if its orientation is the same. Therefore, it is necessary in this case to use rotation invariance techniques. In the related work section, we will see that the state-of-the-art in the studies of extraction of particle characteristics mainly involves spherical particles, which does not need to apply techniques to avoid problems of rotation variation. Also, most of these studies use similar neural network techniques applied in this work for pattern recognition, such as Multi-Layer Perceptron, Radial Basis Function, Deep Learning, and dimensional reduction such as Principal Component Analysis and Autoencoders. However, we did not find related works that use techniques to extract characteristics of prismatic or non-spherical particles from light scattering patterns.

From the images obtained from the light scattering patterns of the particles, it is desirable to extract some characteristics, such as aspect ratio, particle shape and size. However, the feature of the crystal that we want to determine in this work is the projected particle size as seen by incident light from 2DLS pattern. This is an impor-

tant parameter because if we draw the projected size mean from the set of orientations of each particle, we thus obtain an indication of the real particle size, which is necessary as an input parameter for several purposes in other segments of studies, such as radiative transfer calculations that feed climate models (HESSE et al., 2009).

In other words, the main problem involving this work is about the extraction of atmospheric particles features through pattern images. Therefore, it is a reversal problem of a physical process where atmospheric particles are captured by instruments such as SID and that project images by 2DLS which are used to extract characteristics of these particles.

Thus, in this current work, we propose computational models using neural networks techniques aggregated to other techniques of data treatment and rotation invariance of light scattering patterns to be able to predict characteristics of non-spherical particles (especially hexagonal ones) from a database with 2DLS patterns of 162 particles provided by Stopford (2010).

In this study, other techniques could be used to achieve the same goals of pattern recognition, such as genetic algorithms and Bayesian networks. However, we decided to use computational models based on neural network techniques because we found that similar works which try to solve problems in the pattern recognition space with acceptable computational time and quality of the recognition, use the same techniques for different purposes. We found that using a parametrised neural network model and a correct pre-processing stage, we could extract the characteristics of the studied particles in a reasonable time and with satisfactory accuracy that is acceptable in atmospheric science and correlated studies.

The particle patterns dataset used in this work in the training and test stages was kindly provided by the collaboration group composed by Dr. Yi Sun, Dr. Evelyn Hessel, Dr. Neil Davey and Dr. Christopher Stopford, member of the Science and Technology Research Institute of University of Hertfordshire (UH), from UK, as part of an informal collaboration project between UH and UFSC L3C laboratory.

1.2 OBJECTIVES

The aim of this work is to propose and compare four computational models using neural networks (NN) that can effectively predict the projected size of atmospheric particles. To achieve the aim of this

work, the following objectives must be fullfilled:

- Propose an efficient computational model using neural network techniques that can predict projected particle sizes through 2DLS patterns;

- Test and compare proposed models through experiments with the 2DLS particle pattern dataset;

The following tasks must be performed to achieve the objectives of this thesis:

- Understand the characteristics of 2DLS particle patterns and applying image treatment techniques such as normalisation and matrix manipulation to process these images;

- Applying dimensional reduction techniques using Principal Component Analysis (PCA) and Autoencoders;

- Applying rotation invariance techniques using Fast Fourier Transform (FFT);

- Analyse state-of-the-art in the area of particle pattern recognition using Machine Learning techniques;

- Developing computation models with Multi-Layer Perceptron (MLP) neural network techniques using Feed-Forward for classification and pattern recognition purpose;

- Developing computation models using Radial Basis Function (RBF) neural network techniques for pattern recognition purpose;

- Developing computation models applying Deep Learning techniques and Autoencoders neural networks for dimensional reduction and pattern recognition purpose.

1.3 CONTRIBUTIONS

This work focus in contributing, by proposing an effective model, to the study of patterns recognition images in the field of computer vision and climate models currently used by atmospheric science.

1.4 ORGANIZATION OF THE THESIS

This thesis is organized as follows:

- **Chapter 2 - Related Works:** This chapter presents a state-of-the-art literature review on the techniques of predicting atmospheric particle characteristics from 2DLS patterns using Machine Learning
- **Chapter 3 - Methodology and Development:** This chapter details the characteristics of the four models developed to achieve the objectives mentioned in subsection 1.2, as well the hypotheses that led us to develop more than one model to achieve these objectives. It also discusses the dataset used for neural network training and tests and the problems faced with the rotation variation of the particle image patterns.
- **Chapter 4 - Experiments and Results:** This chapter describes and discusses the experiments performed with the four models developed and their results.
- **Chapter 5 - Conclusion:** This final chapter presents the final discussion about the models developed and some proposals for future work.

2 RELATED WORKS

As the inverse light scattering problem occurs in various areas such as Geology, Biology, Astrophysics and Engineering among others, there are some related works which use different neural networks models to extract information from light scattering for distinct purposes that we presented here.

In Ulanowski et al. (1998), the approach used was a Radial Basis Function (RBF) neural network to predict particle size from Multiangle Dynamic Light Scattering (MDLS) patterns. The results presented in Ulanowski et al. (1998) were satisfactory, however the work is focused on spherical particles and does not use rotational invariance techniques in the diffraction patterns (ULANOWSKI et al., 1998) to prevent the problem of rotation variation of patterns as we applied in the present study.

Kaye, Hirst e Wang-Thomas (1997) investigated a different type of elements for classification using light scattering, compared to current work: the applicability of a RBF neural network for the classification of potentially hazardous airborne fibers based on their light scattering pattern. In this work, 99% of correct classification was obtained in real time if appropriate selection of the training template data has been made by a microscopist.

In the image processing area, El-Bakry Hazem M.Mastorakis (2007) used a normalised neural network for fast pattern detection for a given image by decomposition in sub-images. Like in the work described in this study (see chapter 3), they use a Multi-Layer Perceptron (MLP) neural network for pattern recognition and a Fast Fourier Transform (FFT) method for image preprocessing. However, they work focused on improving the speed of the pattern recognition process by using images in the frequency domain and not in the space domain, as this present study does.

In Beaudoin N.Beauchemin (2002), the authors discuss different methods for image processing using Fourier Transform. Their work showed that they could obtain the transformation results in frequency domain in a fast and accurate process, showing that the Fourier transform is a good method for image processing that can be applied in different fields of study.

Another related work that presented good results using measurements by MDLS and NN is given by Gugliotta et al. (2009) which focuses on the prediction of the particle size distribution in polymer

latexes using General Regression Neural Network (GRNN), which is a particular case of an RBF network (GUGLIOTTA et al., 2009). The author also demonstrates that despite having used a simple approach, it has some advantages like: it is a tool besides being simple, it is fast and robust it does not require any specific method for numerical inversion; it is not necessary to adjust parameters, thus allowing the development of an automatic estimation procedure. Although not show in their work, the NN used also proved to be insensitive to the noise patterns in the measurements.

In Guardani, Onimaru e Crespo (2001) uses a similar Feed Forward neural networks of this current work and presents good results, however their work is focused on predicting the particle size distribution as in Gugliotta et al. (2009), particularly of crystals in precipitation processes with antisolvents.

An interesting study by Riefler e Wriedt (2008) makes a comparison of inverse algorithms and methods to predict the size of spherical particles at intervals between 0.1-10μm and 0.05-1μm using Mie scattering patterns (MIE, 1908), which is based on Maxwell's equations (GRIFFITHS, 1999). The main focus of interest was to find out which inversion method is optimal. Riefler e Wriedt (2008) argue that it depends on particle sizes, where they conclude that the best results are often provided by iterative methods. However, Riefler and Wriedt demonstrated that there is no reasonable method that covers all the cases studied. They also demonstrated that the problem must be tested with different methods to see which are the best methods for each case study, even for homogeneous and spherical particles.

Regarding techniques of recognition and classification of patterns from images of other natures using RBF, there are several works with good results for the proposed objectives of each work (ER et al., 2002; BHOWMIK et al., 2009; GUGLIOTTA et al., 2009; KAYE; HIRST; WANG-THOMAS, 1997; GEORGE, 2007). Different from the scope of the recognition of atmospheric particle patterns from 2DLS data entry, some of these works analyse techniques that can be used or adapted for the purpose of the current work.

The most common problem was found in the field of human faces recognition or recognition of images from objects visible to the naked eye. Er et al. (2002) used PCA techniques for extraction of faces features and RBF as a classification technique for face recognition allowing error rates below 4%. However, Er et al. (2002) realized that the technique employed in their work would have better results if the size of the features extracted by the PCA was around 25 to 30 features.

Below or above this numbers, according to the author, would makes learning more difficult or would have lack of information for less features, or even could obtain irrelevant information such as noise, which could disrupt the network in training and testing steps.

Bhowmik et al. (2009) used RBF and PCA techniques to recognize patterns from the fusion between visual and thermal images, seeking to combine both type of images, in order to be able to recognize the patterns and classify with greater precision. After the fusion of images as weighted sum, the images are projected into eigenspace and classified using RBF (BHOWMIK et al., 2009). The efficiency rate obtained in this work was 96%.

In Hodgson (2000), it was used Genetic Algorithm techniques to recognize characteristics of small particles through Light Scattering patterns. This study obtained good results, even in cases where the noise level in the data was relative high. However, the studied particles are spherical, which makes it unnecessary to use invariant rotation techniques.

In the same way that many studies are still focused on the recognition of face patterns, naked eye objects and similar objects using RBF as a Machine Learning technique, the state of the art in the field of Deep Learning also consists of works focused more on these goals. Although 2DLS patterns of atmospheric particles have peculiar characteristics, different from faces, visible objects and others, some techniques that use Deep Learning concepts can be adapted to the current work.

A pioneering work using dimensional reduction technique and extraction of image characteristics through concepts of Deep Learning and Deep Belief Network (DBN) was the study of Hinton e Salakhutdinov (2006). This study proposes a method where two unsupervised Autoencoder networks in sequence are used, taking advantage of the hidden layer output of the previous Autoencoder in order to extract the maximum number of features from peoples faces images, so considerably reducing the size of the input. Hinton and Salakhutdinov also demonstrate that this technique can be more effective than PCA in extracting image features.

Another interesting work involving dimensional reduction which relied on Hinton and Salakhutdinov's work was in Wang et al. (2015), which also uses Autoencoders method and compared it with PCA and other dimensional reduction techniques applied in face recognition field, written characters and other studies of pattern recognition. WANG et al. found that, in some cases, Autoencoders not only reduce dimensionally but also detect repetitive structures, which may be an interesting pro-

perty that can be applied to various purposes in the computer vision field.

In Zhang, Li e Zhu (2015), the study involves the recognition of face images using Deep Learning based on Sparse Autoencoders. This work uses two Sparse Autoencoders in the middle layer and a Softmax Classifier network in the output layer, obtaining reasonable results with approximately 94% of accuracy. However, ZHANG; LI; ZHU points out that the speed of the Deep Learning technique applied for face recognition was extremely slow, which makes the application of this technique very restricted. The author also emphasises that the main focus of the Deep Learning theory is finding the best quantity and structure of layers, as well as the amount of neurons used on each layer.

Salawu (2015) also aimed to extract the particles size by studying the same particle dataset. SALAWU used different methods of neural networks such as Support Vector Regression (SVR) with rotation invariance given by Zernike Moments (ZM) obtaining results with an accuracy of 99%. As discussed by SALAWU, this high accuracy only occurs when the ZM rotation is around 20 degrees. However, in other degrees of rotation, the accuracy decreases considerably.

As presented in this chapter, there are some studies that use distinct computational intelligence techniques for particle regression, which are applied for features that diverge from ones studied in this work, such as some techniques of patterns analysis and computational process (GUARDANI; ONIMARU; CRESPO, 2001; GUGLIOTTA et al., 2009; HODGSON, 2000; RIEFLER; WRIEDT, 2008; ULANOWSKI et al., 1998). In the case of spherical particles, there is no evidence of the use of rotation invariance techniques, since spherical particles have the same pattern of light scattering at different angles of rotation (ULANOWSKI et al., 1998). In some works, different instruments are also used to capture light scattering information from spherical particles (GUARDANI; ONIMARU; CRESPO, 2001; GUGLIOTTA et al., 2009; KAYE; HIRST; WANG-THOMAS, 1997; ULANOWSKI et al., 1998). Table 1 shows a summary of these related works presenting the type of element analysed, their goals, Main Techniques (MT) applied to achieve them and if uses Rotation Invariance (RI) technique.

Table 1: Summary of related works including their goals and methods applied on each of them

AUTHORS	*ELEMENT*	*GOAL*	*MT*	*RI*	*DETAILS*
KAYE et al., 1997	Airborne fibers	Classification	RBF	No	Considerably difference between classes of fibers
ULANOWSKI et al., 1998	Spherical particle	Particle Size	RBF	No	Only spherical particle
HODGSON, 2000	Spherical particle	Characteristics	GA	No	Use Light Scattering patterns
GUARDANI; ONIMARU; CRESPO, 2001	Spherical particle	Size distribution	MLP	No	Precipitation processes with antisolvents
BEAUDOIN; BEAUCHEMIN, 2002	Methods of image processing	Fast processing	FFT	No	Study of FFT
MENG et al, 2002	Face/ objects	Recognition	RBF with PCA	No	Better results with PCA was around 25 to 30 features
HINTON, 2006	Face/ objects	Extraction of image characteristics	Auto encoders	No	More effective than PCA
EL-BAKRY; MASTORAKIS, 2007	Face/ objects	Recognition	MLP with FFT	No	Decomposition in sub-images
RIEFLER; WRIEDT, 2008	Spherical particle	Particle Size	Ten different algorithms	No	Comparison of inverse algorithms and methods

BHOWMIK et al., 2009	Face/ objects	Recognition	RBF with PCA	No	Fusion between visual and thermal images
GUGLIOTTA et al., 2009	Spherical particle	Size distribution	GRNN	No	Particles in polymer latexes
WANG et al., 2015	Face/ objects/ written characters	Recognition	Auto enco-ders	No	Compare with other techniques
ZHANG et al., 2015	Face/ objects	Recognition	Sparse Auto enco-ders	No	Technique used very slow
SALAWU, 2015	Atmosferic Prismatic Particles	Particle Size	SVR and ZM	Yes	Use Light Scattering patterns and good accu-racy only in 20 degress of rotation with ZM

As shown in Table 1, the study which approximate more with the current work in terms of study space, such as element of study, goal and details, is the Salawu (2015) work. This current study applies similar methods, such as normalisation and dimensional reduction, in some developed models to understand the problem faced by them. We also tried to continue their study using different approaches to solve the rotation invariance problem in all possible angles. Besides, we developed a different machine learning method to achieve the main goal in real and practical situations.

3 METHODOLOGY AND DEVELOPMENT

After presenting a summary of computational techniques applied for particle features recognition (see chapter 2), in this chapter we present four computational models to predict projected particle size. For each proposed model, we describe their methodologies and the whole process of machine learning involved, since data preparation (composed by data normalisation, Z-score, split of the particle dataset between training and test sets and random rotation of images), dimensional reduction and their application in neural network models. The four models are composed by four steps as shown in Figure 5.

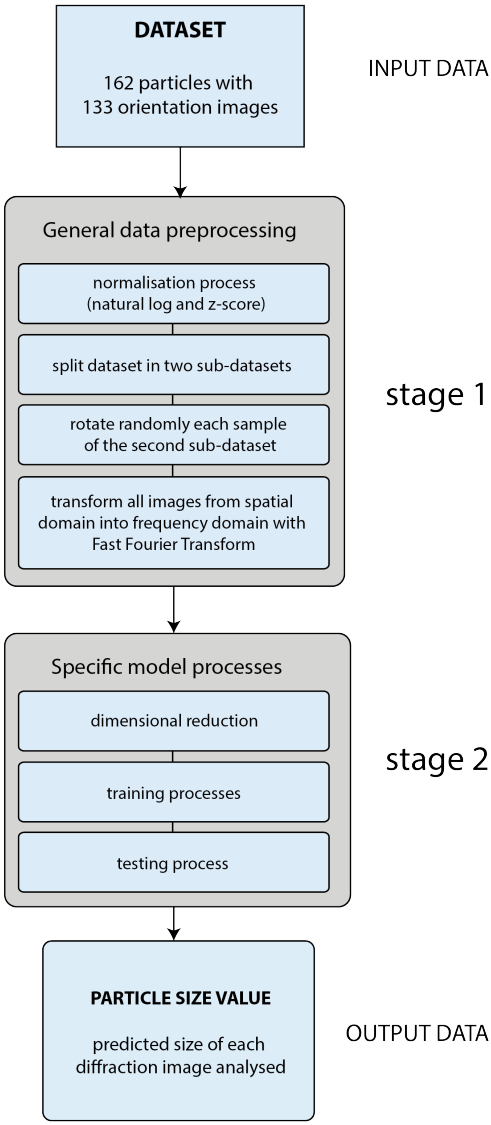


Figure 5: Diagram of general processes for all models.

All four models described here follow the same two general stages with a flow sequential process such as shown in the Figure 5 diagram. The dataset analysed, the first stage and the output data desired from the test phase processed is the same for all models, where the dataset is the content of image patterns of analysed particles; the first stage is the general preprocessing with normalization, splitting into two subdatasets for training and testing process, and image transformation from spatial domain into frequency domain for each pattern. Despite the second stage, all models have equivalent processes, such as dimensional reduction, training and test process with neural networks, where each model uses different techniques (as shown in section 3.6), i.e. different dimensional reduction techniques, different training and testing process. After that, all models have the same output data type: projected particle size of each pattern received in the test process.

To explain in more details each stage and their main steps, we divided this chapter in follow sections. In section 3.1, we present the characteristics of the dataset and the particles images used in this work. In section 3.2, we present the first step on stage 1 (as shown in Figure 5) which is the normalisation processes that we chose. In 3.3 section, we explain the process of splitting the dataset into two sub-datasets for training and test respectively, where each pattern of test dataset is randomly rotated. In section 3.4, we present the problem of the invariant rotation of particles which are captured by 2DLS pattern and the technique used to solve this problem, transforming each image from the spatial domain into the frequency domain. And finally, in section 3.5 and 3.6, we present the details of stage 2 of Figure 5. In section 3.5, we present the dimensionality reduction processes applied in this work and in section 3.6 we explain in more details the four computational models proposed since the preprocessing steps, the dimensional reduction and neural networks techniques that were developed for the proposed objectives, as well as the limitations that we encounter throughout the process and the reasons for developing different models to reach the main goal.

3.1 DATASET

As shown in Figure 5, the first step of the process is understanding the dataset. The particle dataset used in this work was generated by a simulator (STOPFORD, 2010) obtained from the Ray Tracing with Diffraction on Facets (RTDF) model (HESSE et al., 2009), which is a

hybrid model combining trace of light rays with approximation of optical physics. The dataset is composed of 162 particles and each particle has 133 orientations. For each orientation of a particle, a 2DLS pattern is produced. These are computed as intensity profile in spherical polar coordinates with one degree bins. The elevation angle is measured from the incidence direction of the illuminating laser beam and it is recorded between 6° and 25°; the full range of azimuth angles between 0° and 360° is considered. The combination of the elevation-azimuth angles generates pattern images with a resolution of $7,200$ pixels. (see Figure 6).

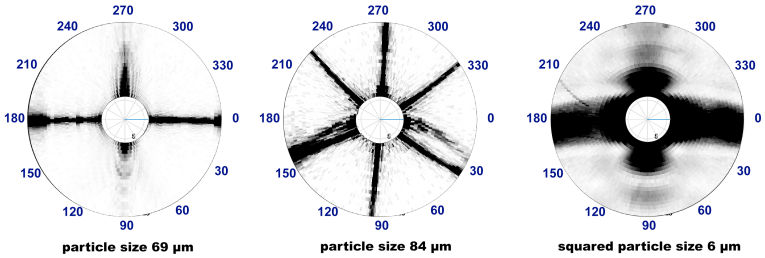


Figure 6: Examples of diffraction patterns in disk format showing the intensity values with some particles of different sizes (in μm).

For all training and testing phases of the models presented we use this dataset of 21,546 patterns (162 particles x 133 orientations), which are split into two sets: 70% of the patterns are used for the training process and 30% for the testing process. In the testing process, we rotate the patterns randomly between 1°and 360°, trying to simulate the real conditions during particle observation in the atmosphere. The patterns analysed by the neural networks (NN) are intensity values of the diffraction images, which are then transformed and normalised by different computational techniques as described in this work. With this information, the NN analyses the size value of each orientation as training output value. Thus, the NN is capable of training and, after that, test with new patterns to predict projected particle sizes.

In section 3.6 we discuss in more details the models developed using different NNs that are able to receive pre-processed samples as input and show as output the value of the size prediction of each sample.

3.1.1 Dataset correlations

To know more about the behavior of the dataset and estimate if using machine learning approaches is a good way to create a computational model for size value prediction from 2DLS patterns, we use the descriptive statistics from Salawu (2015) to analyse the details of the dataset, because is the same dataset used in both works. However, to understand if each image pattern has a correlation with their real size value, we applied a correlation method using Pearson's correlation in the dataset and extract the mean of the correlation and p-value variables, as shown in Table 2.

Table 2: Statistics from dataset and correlation means between size and image patterns

Nº samples	Correlation mean	Significance (alpha)	P-value mean
21,546	-0.2942	0.05	0.0028

As shown in Table 2, the correlation mean between size value and their image patterns is slightly negative, that is, inverse correlation. The correlation mean is the mean of all the correlations between each pattern image and its projected particle size. The p-value mean demonstrates that the null hypothesis of this correlation can be rejected, that is, the p-value is small, less than 0.05 of significance, then the correlation mean between intensities of image patterns and their size values is different from zero, which means that the hypothesis of exist some correlation is true, however, perhaps a weak correlation between input and output data. Thus, the training and testing process using neural networks may encounter difficulties in the direct correlation between the intensity values and respective size values of each sample and only find correlation through the image shapes of the patterns and other internal features of each pattern instead of being only the value of the intensities of each sample pixels.

3.2 NORMALISATION

We applied two techniques for normalisation in sequence: natural log transformation and Z-score scaling. The principal purpose of natural log in this case is help to reduce the range in the features to

a more reasonable scale that improves the performance of the neural networks. With Z-score, we can create a standard distribution with shifting the mean of the feature to zero and makes its standard deviation equal to one. We have tried other techniques before: others log scaling and process without z-score. However, these two techniques, natural log and Z-score, together and in sequence presented the best performance in the test process (SALAWU, 2015).

3.3 SPLIT AND RANDOM ROTATE

In this two steps on stage 1, the particles dataset is split in two datasets: the first one with 70% of the patterns is used for training process and the second one with 30% is used for testing process. After that, all the patterns from second dataset with 30% for testing process are randomly rotated for simulate real condictions, when the 2DLS instruments receive the particles with unknow rotation.

3.4 ROTATION INVARIANCE

In the last step on stage 1, we use the Fast Fourier Transform (FFT) method in order to make sure that patterns which only differ by a rotation around the axis corresponding to the direction of the laser beam are recognised as identical, i.e., that generate only one pattern from the image analysis regardless its rotation.

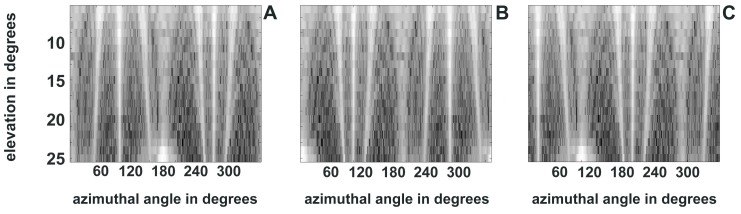


Figure 7: Diffraction pattern images of the same particle with different rotations. Plot A shows the original image of a particle before its rotation. Plots B and C show two different diffraction images after random rotation by 189° and 286° respectively. The x-axis represents the azimuthal angle (ranges of values between 0° and 360°) and the y-axis represents the elevation angle (between 6° and 25°).

With the FFT method, we are able to convert the pattern from image values to frequency values (BEAUDOIN N.BEAUCHEMIN, 2002; EL-BAKRY HAZEM M.MASTORAKIS, 2007), making the image rotation-invariant, and from that, we could predict particle size. Without FFT method, the rotated particle presents distinct pattern images to the NN as shown in the plots B and C of Figure 7, which are two different diffraction images from the same particle rotated by 189° and 286° respectively. These images are only displaced by different angles compared to the original image shown in plot A of Figure 7. Thus, although it is the same image pattern, for the NN it appears as a completely different pattern. After transforming each pattern into a set of frequency patterns (one for each elevation angle) to solve the problem of rotation variation, the diffraction pattern images shown in Figure 7 are converted from spatial domain into frequency domain like those shown in Figure 8. Each line in Figure 8 represents a single elevation of each pattern shown in Figure 7, transformed into frequency values by FFT method. Note that three lines are overlapped in the graph, especially at the frequency range from the value zero to the value of 100.

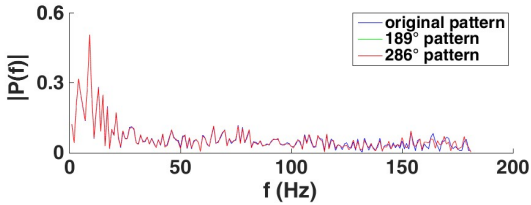


Figure 8: Frequency values of the same 2DLS with different rotations, exemplifying the importance of using the FFT method. The blue line shows the frequency of a pattern before rotation. The green line and red line show the frequency of the same pattern at same elevation, after random rotation by 189° and 286°, respectively. The x-axis shows the pattern frequency (in hertz) and the y-axis represents the frequency amplitude of sinuses or cosines of the pattern.

3.5 FEATURE SELECTION AND DIMENSIONAL REDUCTION

This step is specific for each model applied in this work and belong to stage 2 (see Figure 5). Feature selection is an important approach to reduce the dimensionality of the patterns allowing to be more computationally efficient for machine learning techniques with minimal

or null significant loss. There are a range of feature selection approaches such as Univariate Feature Selection (UFS), Recursive Feature Elimination (RFE) (SALAWU, 2015), Linear Discriminant Analysis (LDA), Isomap and others (WANG et al., 2015). However, we applied two approaches in this work such as Principal Component Analysis as shown in section 3.5.1 and Autoencoders as shown in section 3.5.2. PCA is used because many studies of pattern and particle recognition applied this technique with satisfactory results in some cases. Autoencoders is used because recent studies in other pattern recognition fields had better results than PCA.

3.5.1 Dimensionality Reduction with Principal Component Analysis (PCA)

When the dimension of the input vector is large but the components of the vectors are highly correlated, that is, redundant, it is useful in some situations to reduce the dimension of the input vectors. An effective procedure for performing this operation is the Principal Component Analysis (PCA) method. This method orthogonalizes the components of the input vectors, so that they are uncorrelated with each other. PCA also orders the resulting orthogonal components (principal components) so that those with the largest variation come first, and it eliminates those components that contribute the least to the variation in the data set. In other words, the majority applications of PCA is to transform patterns into a new space and to use lower-dimensional representation from the new space to denote the pattern, reducing the computational complexity and cost (SONG; GUO; MEI, 2010). In this study, we use the first forty components of PCA that explain around 55% of the total variations in that dataset as shown in Figure 9. However, increasing the components amount to use in the process do not make a better considerably difference in prediction performance. Also, using a number with less components decrease the prediction performance. In fact, different amount of components does not even increase the performance prediction. The more the amount get away from forty components, more the performance decrease. This approach, using forty components, we applied in model 1, 2 and 3.

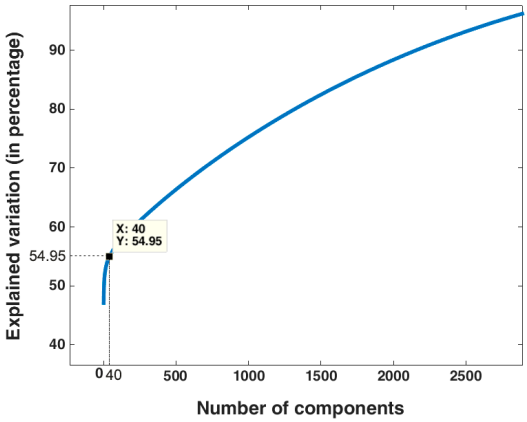


Figure 9: Amount of components that explain the variation in dataset. The y-axis shows the percentage of explained variation and x-axis shows the amount of components necessary to extract the correspondent variation.

3.5.2 Dimensionality reduction with Autoencoders

An autoencoder is a unsupervisioned neural network with minimum three layers which is set the output layer equal to the input layer, that is, the same amount of nodes and targets and the hidden layers usually is set with less nodes. However, when restricting the number of hidden layer nodes to less than the number of original input nodes, we can extract a compressed representation of the input. That is, the desired dimensionality reduction effect is achieved, as illustrated in Figure 10.

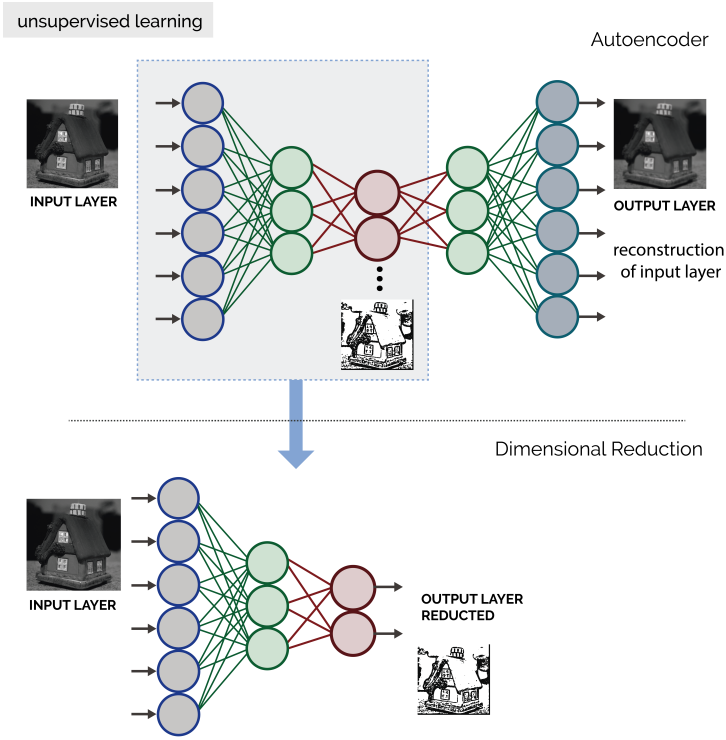


Figure 10: Autoencoder scheme used for dimensional reduction. After unsupervised training, we can use the structure of the autoencoder from the input layer to the hidden layer chosen (usually the layer with fewer nodes) to test and extract a reduced representation of inputs.

In this study, we applied two Autoencoders in sequence for dimensionality reduction. This approach we applied in model 4 and is explained with more details in section 3.6.4.

3.6 PROPOSED MODELS

Once we have determined the training and test set of particle images by 2DLS patterns, we started the development of the first computational model using data processing and NN for projected particle

size prediction. The other models (see section between 3.6.2 and 3.6.4) we developed when we realised the previous model was not have a good performance in the predictions, even for smaller particles, which we explain with more details in the next sections.

3.6.1 Model 1: MLP FF with a Hidden Layer

The proposed NN model is a multi-layer feed-forward neural network with 50 nodes in the hidden layer (see scheme of a MLP Feed Forward Neural Network in Figure 11), which uses a Bayesian regularization backpropagation as training function (MACKAY, 1991). The number of nodes was chosen after analysing the network performance when considering the network model capacity of generalization and interpolation. However, as the training is based on a Bayesian regularization function, the number of nodes in the hidden layer is not an important factor, as this kind of training function minimises the problem of overfitting when a large number of nodes in the hidden layer is used (FORESEE; HAGAN, 1997).

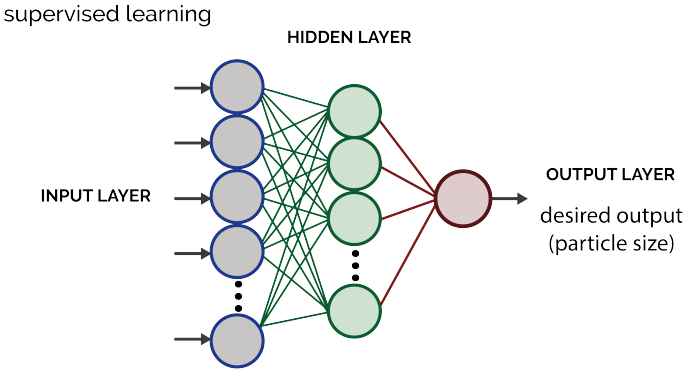


Figure 11: MLP Feedforward Neural Network scheme. This model use 50 nodes in hidden layer and a Bayesian regularization function.

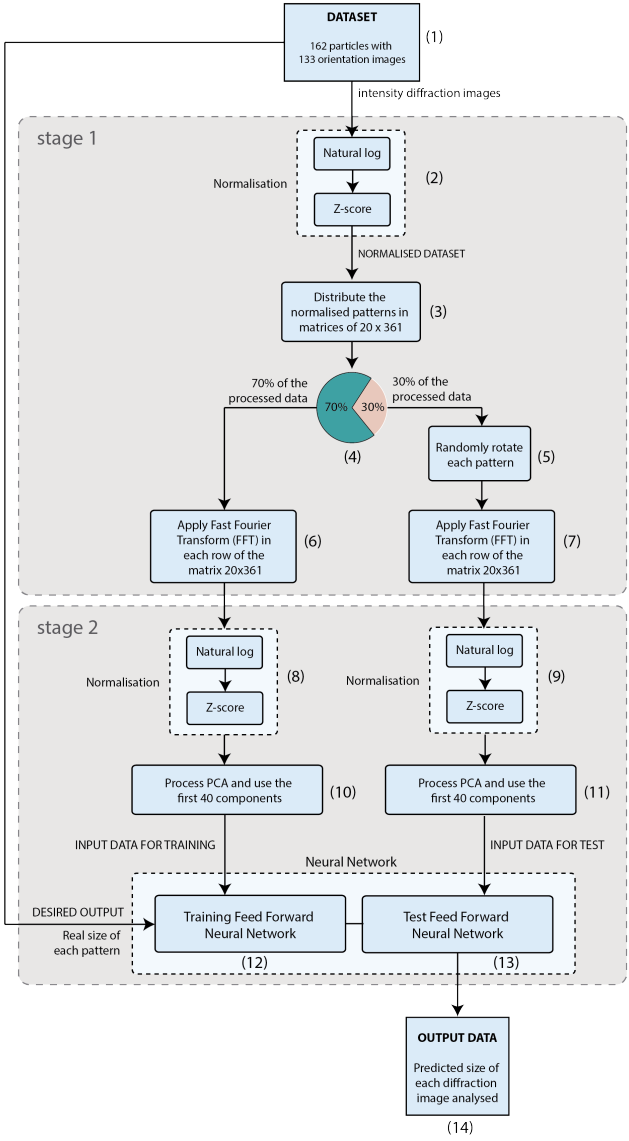


Figure 12: Diagram of prediction process performed by the first proposed model. We can see in this diagram the general data preprocesses (stage 1 of Figure 5) with more details, as shown between step 1 and step 7, and also the specific details of this model in stage 2.

In Figure 12, the prediction process proposed in this model shown the hole process, included the general processes (as shown in Figure 5) and the specific process for this model with more details. The diagram begins considering the particle data set composed of 162 particles and 133 orientation images for each particle (step 1), which generates 21,546 diffraction pattern images.

In first stage, we then normalise the diffraction pattern images using natural log and Z-score measurements (step 2). After that, we distribute each pattern in a matrix of 20 x 361 (step 3), where the 20 rows of the matrix represent elevation angles between 6˚and 25˚, and the 361 columns represent the azimutal angles between 0˚ and 360˚, thus generating a flat image of each orientation - this process is better explained in Section 3.4, where an example of a flat image can be seen on Figure 7.

Next (step 4), we generated two matrices. The first one with 70% of the images processed in the previous steps and the second one with 30% of them. The first matrix is used for training steps and holds the original pattern of the images. The second matrix is used for test steps and holds the randomly rotated patterns (step 5). We apply the FFT method to each line of the both generated matrices (steps 6 and 7), thus generating signal frequency patterns in order to achieve an image without problems of rotation variation (see Figure 8).

After that, we start the stage 2 (between step 8 and 13) with specific processes of this model. In the next steps, the frequency matrices are normalised (steps 8 and 9) because the domain of this values was changed and the PCA process takes longer without applying normalisation in their input. After step 8 and 9, their dimensions are reduced by applying a PCA process (steps 10 and 11). From there, we generated the input data for training and run the training phase of the proposed NN model (step 12). Finally, we use the rotated test patterns, which were obtained from step 5, and applied the FFT method, normalization and PCA process (step 11) to the input data for the test and run the testing phase of the proposed NN model (step 13), where its output gives the predicted particle sizes (step 14).

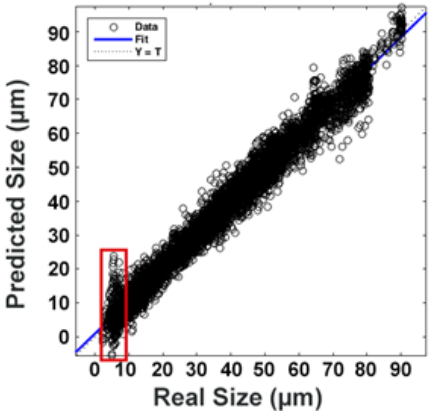


Figure 13: Model 1 plot predicted size against real pattern size. Each datapoint represents one pattern image at one particle orientation. The x-axis shows the projected size of each particle orientation and the y-axis shows the predicted size. The highlighted box (in red) shows that the proposed model has problems in predicting the projected size of these small particles.

In all performed tests using the proposed model, the NN could predict the projected size of the most analysed particles with a good precision. These results are plotted in Figure 13, showing that using 30% of the dataset for the test process, the results represent a normalised mean squared error around 0.007 and the value of determination's coefficient (called "performance"in this work), where 0 for worst performance and 1 for best performance, was around 0.98. However, these results show that it is difficult for the NN to predict particles with small sizes, as shown in the highlighted box of Figure 13, where the predictions of particles between 3 and 10 μm size are plotted.

This behavior occurs most often when the model is submitted to the testing phase, especially when certain specific patterns of some particles are placed in the test set as shown in Figure 14 which is a plot with only a small particle (5.3 μm) and all the 133 orientations predicted and extracted from the prediction plot of the Figure 13.

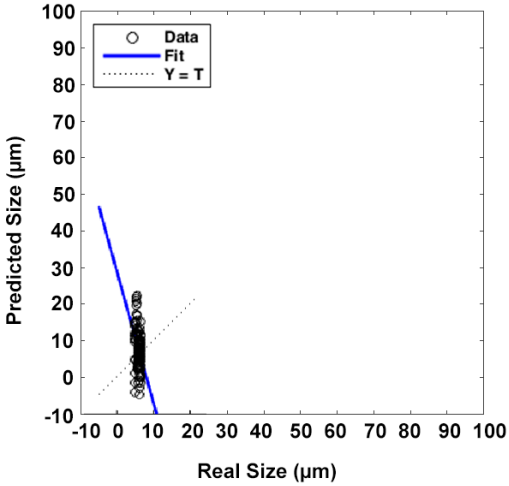


Figure 14: Model 1 plot predicted size against real pattern size of a squared small particle. Each datapoint represents one pattern image at one particle orientation of a small particle with 5.3μm. This plot with only this particle was extracted from the plot of Figure 13. The x-axis shows the projected size of each orientation and the y-axis shows the predicted size. We can see the predicted size range is much bigger than real size range.

It is well known that NNs are universal approximators which would mean in this case that a reasonably parameterized NN should be able to get a small error, but this issue shown in Figure 13 and 14 is indeed relevant in this work because for the application in atmospheric science it is important to decrease the error for smaller particles to a similar values as for larger particles.

In order to investigate the reasons for the model did not have a satisfactory performance for small size particles, we have carried out new experiments, as described below, which we have found that the main problem in this first model with NN approach was to distinguish particles with very dfferent sizes but similar diffraction patterns. (as shown Figure 13).

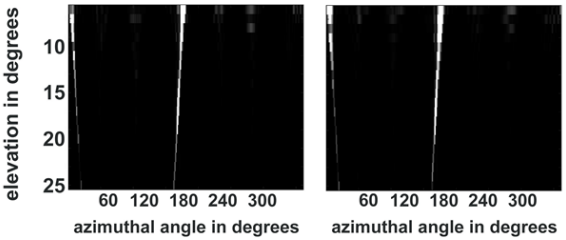


Figure 15: 2DLS patterns from different size particles with the same aspect ratio (equal to 3) and same orientation (Alpha = 0 deg, Beta = 35 deg e Gamma = 15 deg). The left image shows a diffraction pattern from a large particle of 61.4 μm size. The right image shows a diffraction pattern from a small particle with size of 46.6 μm. For both patterns, the NN had problems to predict the actual size of each particle. Note that the diffraction pattern is slightly more narrow for the larger particle, which is expected from the diffraction theory.

In Figure 15 we analysed two different particles of different lengths, diameters and sizes, but which have very similar patterns due to the similar aspect ratio and orientation of the particles. For both patterns, it is difficult for the proposed model to predict the projected size of each particle.

It has also been identified that this generalization problem occurs more frequently in some particles than others and especially with smaller particles. Figure 16 shows a histogram of the total number of orientations for some particles considered "problematic", which means the particles with an Mean Squared Error is higher than 20%. It is important to note in this Figure 16 that the particles which have the largest number of problematic orientations are small (i.e., have small length and diameter).

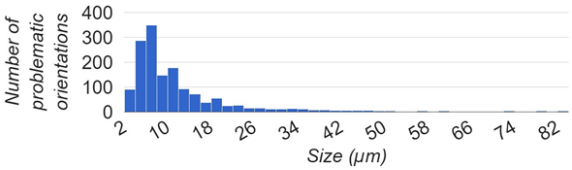


Figure 16: Histogram with the number of problematic particle orientations. The x-axis shows the real size of each pattern and the y-axis shows the amount of problematic orientations. It should be noted that most problem particles are small.

Another interesting point found when analysing the patterns of the problematic particles was that many patterns have the same beta angle as shown in Figure 17.

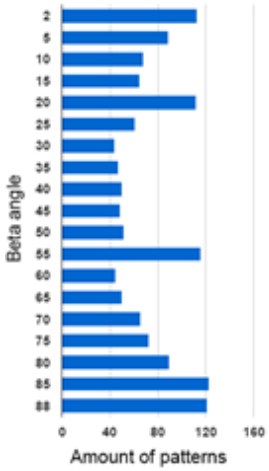


Figure 17: Problematic patterns that have similar beta angles. This histogram shows the number of problematic patterns (presented in the x-axis) for each beta angle (presented in the y-axis).

The reason for this correlation between the problematic patterns and the beta angle could be that smaller particles interact less with incident light than do the larger ones; in addition the light scattered by small particles is more widely spread out. Ultimately, this means that there is less information recorded in the angular region covered by these patterns, which could explain the difficulty in recovering size information from smaller particles.

In a new experiment, the NN model used in the previous experiments was trained and tested using a single small compact particle as input. In this experiment, the network performs well as shown in Figure 18.

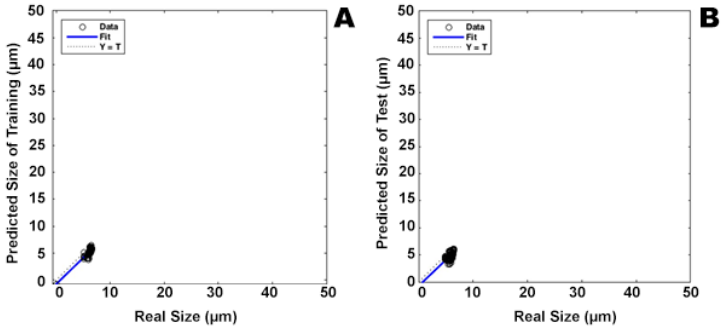


Figure 18: Training and test plot prediction of a unique and small compact particle using Model 1. The x-axis shows the projected particle size and the y-axis shows the size predicted by the neural network for each orientation. Plot A shows the neural network training process using half of the available orientations from this particle as input data. Plot B shows the results of the neural network testing process using the other half of the available orientations from the same particle as input data.

From the successful results obtained from the last experiment, we have decided to develop a new model which used distinct neural networks for different types of particles. This model is described in the next section 3.6.2.

3.6.2 Model 2: Two Multi-Layer Feed-Forward Neural Networks applied for Regression and One for Classification problems

After realizing that the first model composed by a single Multi-layer feed-forward neural network had problems to generalize some image patterns, specially the smaller ones, we have decided to use two neural networks for prediction, in order to facilitate the training and testing of the NNs when receiving specific inputs separately. So, for this experiment, we splitted the input patterns in two datasets: the first NN receives patterns that were considered "non-problematic" for training and test, and the second NN receives patterns considered "problematic" ones. This model is explained in the diagram presented in Figure 19.

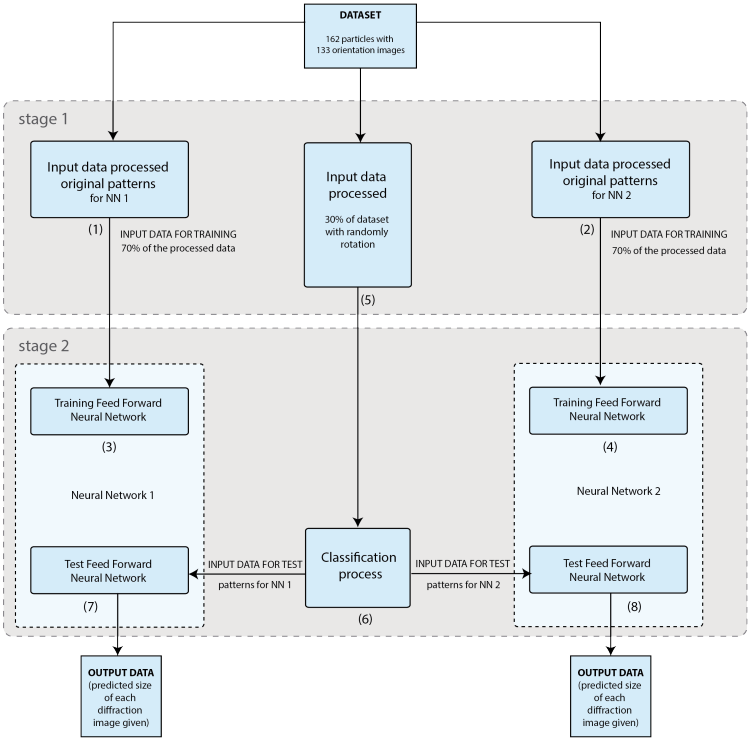


Figure 19: Diagram of the model 2 which uses two neural networks for prediction and one for classification.

In Figure 19, we can see how we structured the model to classify, train and test using two distinct networks. The particle image patterns for training and testing sets are processed ising the same steps as in the model 1 (presented in section 3.4.1.), including normalization techniques, Fourier transform for invariant rotation and dimensional reduction using PCA (inside steps 1, 2 and 5 of the diagram presented in Figure 19).

For the testing set, the image patterns are randomily rotated just before the Fourier transform, and then the PCA process is started (step 5). In order to train (steps 3 and 4) and test (steps 7 and 8) the dataset with two neural networks (NN1 and NN2), we first made a list of orientations that have prediction with error above 20% (i.e., problematic orientations) when training and testing using all samples.

NN1 is a supervised neural network with the orientations which were not in the list of problematic orientations; and NN2 is a supervised neural network with all orientations in the list of problematic ones. Each network uses 70% of the total dataset for training and 30% for testing. However, to simulate a real situation in the test, we added a step to identify whether the images patterns received for testing could be part of the list with the problematic orientations or not, in order to direct the inputs to the corresponding network (step 6). For that, we used a supervised neural network for classification.

The supervised neural networks used for prediction were similar to model 1: multi-layer feed-forward neural network with 50 nodes in the hidden layer and Bayesian regularization backpropagation as the training function. The neural network used to classify the problematic patterns (i.e, patterns with error over 20% of prediction) and non-problematic ones is a feed forward network with 50 nodes in the hidden layer and a training function using Bayesian regularization backpropagation (FORESEE; HAGAN, 1997).

In this experiment we noticed that if the prediction neural network received similar patterns that have been used for training, the network was be able to predict correctly with a satisfactory performance. Especially if the patterns considered problematic are not in sufficient quantities to have well-demarcated patterns but with sizes very close to them, making the generalization more difficult. However, the model had some difficulties to classify new patterns and allocate them to the correct network. As result, both networks ends up having great difficulties of generalization (see the training and test results in Figure 20).

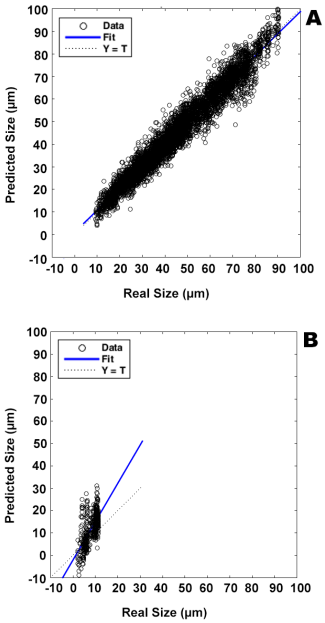


Figure 20: Result of training (plot A) and test (plot B) with a classification network and two prediction networks. In the y-axis, we see a prediction size (in μm). On the x-axis we see the projected real size (in μm).

We then realised that the approach of using MLP feed-forward neural networks to classify and redirect to training and test only image patterns of the same "category"inside the input layer of the standard NN MLP feed-foward for prediction was not solving. The problem ended up being mainly in the difficulty of the classification. We then decided to use another approach that could configure the network and its neurons more independently and selectively between them, in order to minimize the problem of particles considered problematic and the difficulty of predicting small particles.

3.6.3 Model 3: Radial Basis Function (RBF) Network

For this new set of experiments, we developed a model using Radial Basis Function (RBF) network using custom spread values (radius of radial function) of each neuron in order to improve the prediction

of the small particles and the prediction of particles with large differences in size but with little difference in image patterns, as shown in Figure 15 of section 3.6.1.

The input data was processed as in the models 1 and 2, using normalization, Fourier Transform and PCA (see steps 1 to 11 in Figure 21. The difference between this model and the models 1 and 2 was that the training and test processes used RBF tecniques as shown in the steps 12 and 13 in Figure 21).

The RBF approach uses clusterizing techniques in order to each neuron has the activation through Gaussian function according to the approximation of the image pattern outputs that we desire. This model uses K-means in clusterizing method. Each neuron has a gaussian function activation with custom value of spread. Thus, we can have neurons with more selectivity than others (GEORGE, 2007).

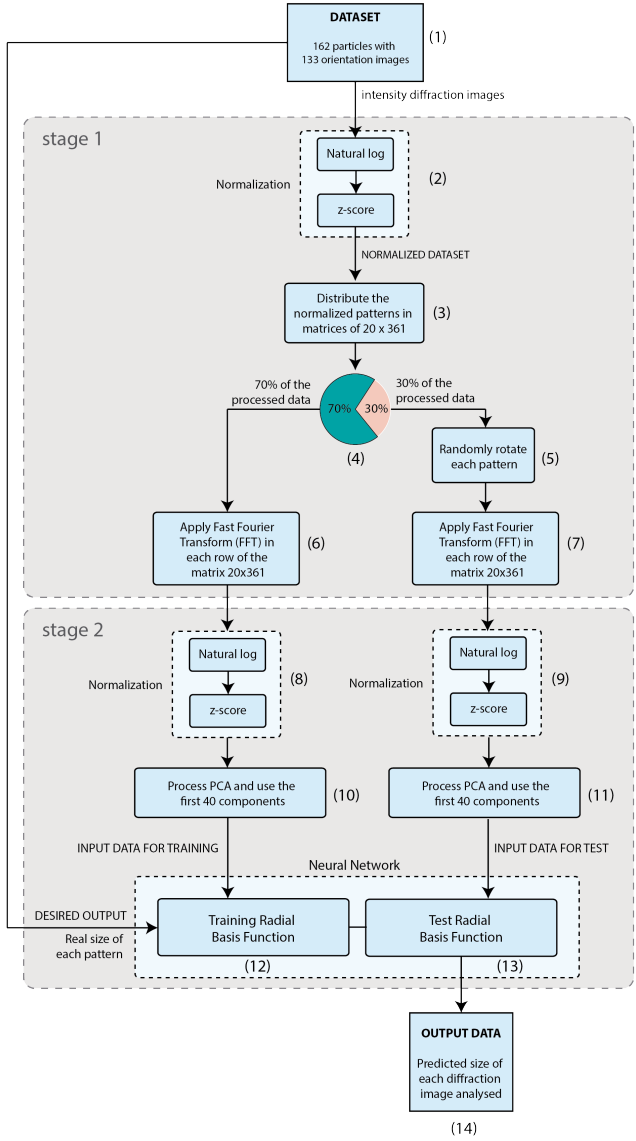


Figure 21: Prediction process diagram with model 3 approach. Note that the pre-process of the samples before training and testing are similar to model 1 (normalization, Fourier Transform, PCA). The main difference is in the architecture of the neural network for training and testing (step 12 and 13), which uses RBF with its characteristics in order to solve the problem of "problematic"particles.

This model obtained better results using a significant amount of neurons: between 4,000 and 5,000 neurons in the hidden layer the results were more satisfactory. Above 5,000 neurons, the model becomes increasingly unfeasible in terms of processing time, as well as a very evident overtraining. However, we obtained slightly better results than the previous models, especially with the smallest particles, considering the entire test dataset (30% of the complete dataset) in a single graph, as we can see in the example of the test process in Figure 22, which was used 5,000 neurons in the hidden layer.

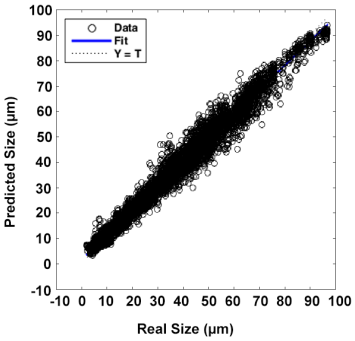


Figure 22: Plot example showing predicted size with projected size (Real size) of test process with model 3. Each data point represents a pattern image in a particle orientation. The x-axis shows the projected size of each orientation of the particles and the y-axis shows the size predicted by the NN model. We see that the performance is around 0.99, but in some regions of sizes we can see a lower precision.

Despite the improvement in the predictions, especially of the small particles (between 0 and 20 μm) there are still regions of particle sizes that the model finds more difficult to predict the correct projected particle size, as we can see the prediction plot example of a small and square particle, with a diameter of 5.3μm in Figure 23.

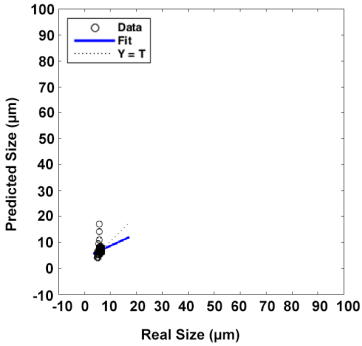


Figure 23: Prediction plot example of a small particle, with 5.3 ?m width and diameter (square particle). We can see that the prediction has a significant prediction error in relation to the real-size values of this particle, which is around 8 μm with little variation, and the prediction has a range between 3 and 18 μm. Thus, the variation of the prediction ends up being relatively high, having a performance, only of this particle, around 0.14

Although we have minimized the problem of predicting small particles (0-20 μm) using this model, we can still see that the prediction of some particles that in this model as in previous models are still not completely satisfactory for the practical use of these models in the field of atmospheric science and in other fields that use the recognition of particle patterns whose accuracy of the prediction needs to be more homogeneous along the different sizes of particles (especially the smaller ones) captured by 2DLS patterns.

Therefore, we began to rethink deeper ways of learning neural networks, but maintaining the current computational viability, as well as rethinking the data processing steps that is processed before beginning the training and testing steps, as we also consider the possibility that in those initial stages some important information of each pattern is being disregarded or suppressed, as it can happen in the principal component analysis (PCA) step where the dimensional reduction employed by this technique can be causing in significant loss of information of each pattern. The dimensional reduction is an important step for the computational viability of the complete process. In this way, we can also verify if the PCA was actually producing noise or suppressing pertinent information from the 2DLS patterns, hindering the prediction process by the neural networks.

3.6.4 Model 4: Deep Learning with Sparse Autoencoders

Based on the analysis of the previous models, we developed a model that uses concepts of Deep Belief Network (DBN) with structure of Autoencoders in order to obtain training more sensitive to the differences of the image patterns in a computationally viable way (BENGIO, 2009), and that it can reduce the dimensionality of the samples in a more efficient way than the PCA without a significant loss of information from each sample that could compromise its prediction (HINTON; SALAKHUTDINOV, 2006; WANG et al., 2015).

An Autoencoder is an unsupervised multi-layer neural network trained with backpropagation by gradient descent and encoding an input x to a representation $e(x)$ and decodes (reconstructs) this representation into an output y (ZHANG; LI; ZHU, 2015). In other words, the dimension of the input x is the same as the output and this output is a reconstruction of the input where the process of coding and decoding extracts the main characteristics of each sample, as we can see in the representation of the functionality of a simple Autoencoder with a unique hidden layer in Figure 24.

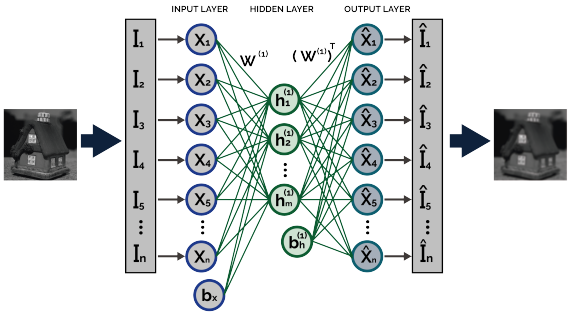


Figure 24: Schematic of an Autoencoder with details. Such as seen in Figure 11 we can notice that the input layer and the output layer have the same amount of neurons. The training is unsupervised. That is, the input is only the samples that are coded and then decoded by adjusting the weights to be able to maintain the maximum characteristics of the original samples.

In Figure 24, we can see the hidden layer has the least amount of neurons and can be multiple layers, with the most central layers having the least amount of neurons and the lateral layers increasing the amount

gradually, such as a bottleneck (SEYYEDSALEHI; SEYYEDSALEHI, 2015).

In this current work we use Sparse Autoencoders, which sparse restrictions are added to autoencoders. The concept of sparsity constraints that we use comes from the notion of the efficiency in which the mammalian visual cortex, due to evolution, transforms the input images in a way that can reduce redundancies and describe the images through primitive structures such as lines, edges and elementary features from sparsity coding (OLSHAUSEN; FIELD, 1997). In computational terms, the sparse restrictions are based on the neurons when are considered to be activated if the output values of neurons close to 1 and they are considered to be inhibited if the output values of neurons close to 0. To do this, we implement some sparse restrictions that we explain below.

The autoencoder encoding and decoding activation function is sigmoid, and as explained earlier, an autoencoder consists of an encoder and a decoder and both can have multiple layers. However, in this current work we use an input layer, one hidden layer and an output layer in each autoencoder. If in the input layer we have a vector x, then the encoder maps the input to another vector z, as shown in Equation 3.1:

$$z^{(1)} = h^{(1)}(W^{(1)}x + b^{(1)}) \tag{3.1}$$

where the superscript (1) represents the first layer, $h^{(1)}$ is the transfer function, $W^{(1)}$ is the weight matrix and $b^{(1)}$ is the bias vector. The decoder maps the encoded z vector back into an estimate of the original input vector, x, as follows:

$$\hat{x} = h^{(2)}(W^{(2)}x + b^2) \tag{3.2}$$

where the superscript (2) represents the second layer, $h^{(2)}$ is the transfer function, $W^{(2)}$ represents the weight matrix and $b^{(2)}$ is the bias vector.

To work with sparsity inside the autoencoders we add regularizers to the cost function. These regularizers are output activation functions of a neuron. The cost function used in sparse autoencoders, with regularizers, is defined in equation 3.3:

$$E = \underbrace{\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K}(x_{kn} - \hat{x}_{kn})^2}_{A} + \underbrace{\lambda * \Omega_{weights}}_{B} + \underbrace{\beta * \Omega_{sparsity}}_{C} \tag{3.3}$$

Note that the term A is a traditional mean squared error equation. The term B is the first regularizer called Weight Regularizer and the term C is the second regularizer, called Sparsity Regularization. The Weight Regularizer is preceded by λ which is the coefficient for the Weight Regularization term and β is the coefficient for the Sparsity Regularization term. The term Weight Regularizer avoids that it is necessary to increase the values of the weights $w^{(1)}$ and decrease the values of $z^{(1)}$ (see Equation 3.1) to let the sparsity regularizer be low (OLSHAUSEN; FIELD, 1997). This term is defined in equation 3.4 by:

$$\Omega_{weights} = \frac{1}{2}\sum_{l}^{L}\sum_{j}^{n}\sum_{i}^{k}(w_{ji}^{(l)})^{2} \tag{3.4}$$

where L is the number of hidden layers, n is the number of samples and k is the number of variables (from a sample) in the training data.

The term Sparsity Regularization of equation 3.3, referring to the cost function, tries to impose a restriction on the sparsity of the output of the hidden layer. Sparsity can be stimulated by the addition of a regularization term that has a high value when the mean activation value $\hat{\rho}_i$ of a neuron i and it is desired value, ρ, are not close in value (OLSHAUSEN; FIELD, 1997). In other words, is a penalty to the error function which will prevent the activations from straying too far from the desired ρ. The Sparsity Regularization term used in this work is the Kullback-Leibler (KL) divergence, as follows:

$$\Omega_{sparsity} = \sum_{i=1}^{D^{(1)}} KL(\rho \parallel \hat{\rho}_i) = \sum_{i=1}^{D^{(1)}} \rho \log\left(\frac{\rho}{\hat{\rho}_i}\right) + (1-\rho)\log\left(\frac{1-\rho}{1-\hat{\rho}_i}\right) \tag{3.5}$$

The Kullback-Leibler divergence is a function to measure the relative entropy, that is, the difference between two distributions, where the value is zero when ρ and $\hat{\rho}_i$ are equal to each other, and become larger as they diverge from one another. Thus, this term must be small so that the cost function is minimized. In general, $KL(\rho||\hat{\rho}_i) \neq KL(\hat{\rho}_i||\rho)$, although it is sometimes referred to as Kullback-Leibler distance, is not exactly a distance (i.e., Euclidean distance) and is an important relative entropy function for neural network techniques, pattern recognition, and information theory (MACKAY, 2002).

The measure of activation of the mean output $\hat{\rho}_i$ of a neuron i is defined in equation 3.6:

$$\hat{\rho}_i = \frac{1}{n}\sum_{j=1}^{n} z_i^{(1)}(x_j) = \frac{1}{n}\sum_{j=1}^{n} h(w_i^{(1)T} x_j + b_i^{(1)}) \tag{3.6}$$

where n is the total number of training samples, $w_i^{(1)T}$ is the ith row of the weight matrix $W^{(1)}$, x_j is the jth training sample and $b_i^{(1)}$ is the ith entry of the bias vector, $b^{(1)}$ (see Figure 24). As told before, a neuron is activated if its output activation value is high, thus is, close to 1. A low output activation value, close to 0, means that the neuron in the hidden layer is activated in response to a small number of the training samples. Adding a term to the cost function that restricts the value of $\hat{\rho}_i$ to be low encourages the Autoencoder to learn a representation where each neuron in the hidden layer is activated for a small number of training samples. That is, each neuron specializes in responding to some characteristic that is only present in a small subset of the training samples.

In this way, we developed a neural network model with an input, two hidden layers and an output layer, where the two hidden layers obtain the characteristics and structure of pre-trained unsupervised autoencoders networks and the third layer is a supervised network with a unique layer which has a linear transfer function, where the cross-entropy performance function as the Mean Squared Error and a training function as Scaled Conjugate Gradient Backpropagation (see Figure 25, 26 and 27). In other words, we introduce concepts of deep learning in each layer of a multi-layer feed-forward neural network (WANG et al., 2015; ZHANG; LI; ZHU, 2015).

The purpose of using this technique is to naturally decompose the problem into sub-problems associated with different levels of abstraction. An adequately modeled and properly pre-processed unsupervised learning algorithm can extract obvious input information (COATES; LEE; NG, 2011). But often, some characteristics become evident only through a distributed representation, that is, only through the layers of a multi-layer network and resulting from more than one sample. Thus, an unsupervised learning algorithm with a single layer could extract such characteristics, however, due to the limited capacity of this layer, the resources extracted from it can be seen as low-level characteristics. If we introduce a second layer soon after the first and use the output of the first layer as input of the second one, this could extract resources of slightly higher level (BENGIO, 2009). Thus, it is possible to detect higher-level abstractions that characterize the input and that would be less possible to emerge in a traditional MLP feed-forward network

architecture with a single hidden layer.

Figure 25 shows the diagram of all the steps of this model. Figure 26 shows a more detailed diagram of the Autoencoders that have been trained separately, which are later reconstructed into a single multi-layer neural network. Figure 20 shows the multi-layer neural network model reconstructed from the three networks shown in Figure 25 and 26.

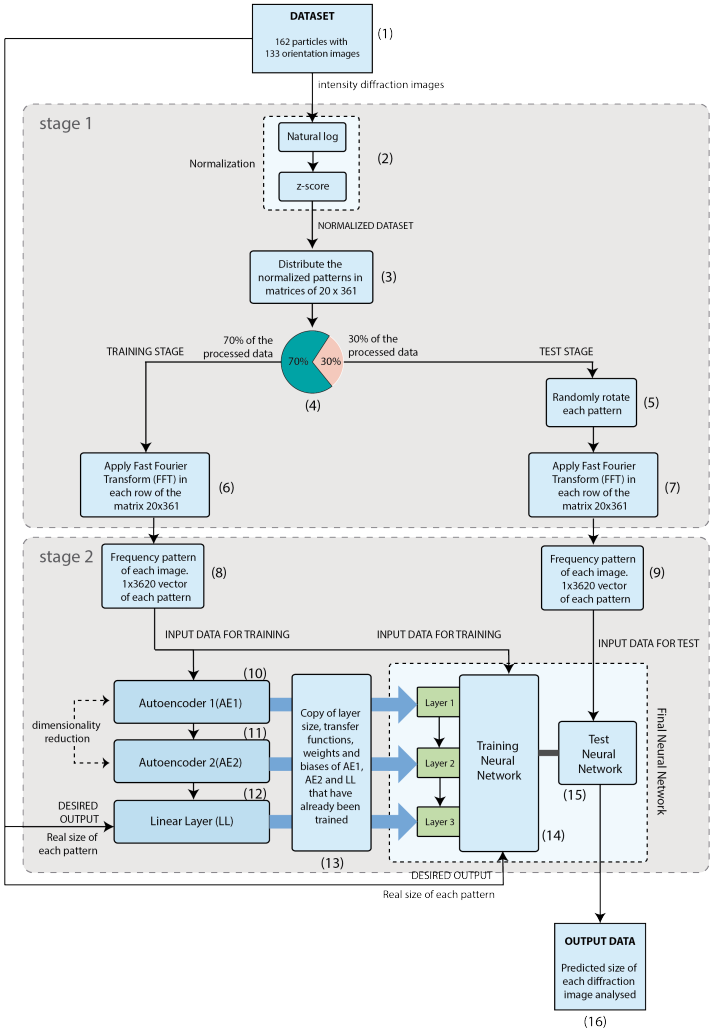


Figure 25: Diagram of the complete model 4 that use concepts of Deep Learning with Sparse Autoencoders as training stage and test with neural networks. In this diagram we can see the steps from the preprocessing (normalization and Fourier transform) to the following steps of the Autoencoders training and the network with linear transfer function, as well as the construction of a deep learning network from the copy of the structures of each Autoencoder, without its layers of outputs (see Figure 26 and 27) in order to do a new training and, finally, the final test with the samples of the test dataset.

As we can see in Figure 25, the model starts the preprocessing of the dataset samples in the same way as the previous models, from normalization to the FFT (to transform into frequency pattern) as we can see from step 1 to 9. These steps are applied to each sample in both the training and test dataset. From there, the differences of the other models developed in this current work begin: we did not use the dimensional reduction with PCA after FFT. The generated frequency patterns of each sample are placed directly as the input of the first Autoencoder (step 10) to perform a pre-training and a dimensional reduction (steps 10 through 14 of Figure 25 can be seen in more detail in Figure 26 and 27).

Then the output generated from the training by the hidden layer of Autoencoder 1 becomes the training input of Autoencoder 2 (step 11) and thus a new pre-training is performed by Autoencoder 2 and the output generated by the hidden layer of the Autoencoder 2 becomes the input to the supervised network with linear transfer function (step 12). As it is a supervised network, the training of this network is made with the projected particle size of each sample. Thus, the desired output of this network is the projected particle size of each sample presented to this last network.

After finalizing the pre-training of the three networks, a final feed-forward network with two hidden layers and an output layer (step 14) is constructed: In the first hidden layer we created the same amount of neurons of the hidden layer of Autencoder 1 ($h^{(1)}$). We follow the same concept in the second layer with the second Autoencoder and its hidden layer ($h^{(2)}$). Then we put the adjusted weights and bias of the hidden layers of the two Autoencoders, as well as the transfer function (sigmoid), performance function (mean square error) and training function (scaled conjugate) in the respective layers of the final network. In the output layer we put a single output neuron with the adjusted weights, bias and the linear transfer function of the previously created network.

Next, we re-train with this new multi-layer supervised network with the training dataset and the projected particle sizes of each sample. The interesting thing is that as each layer already has its weights previously adjusted by the pre-training stage performed in Autoencoders and in the network with linear transfer function, this final network, with all layers in a single network, can obtain a training process faster than if the first training (with the original input received by the Autoencoder 1) was done directly in that final network (BENGIO, 2009), which often becomes computationally infeasible. Finally, the final network is

tested using the test dataset (step 15).

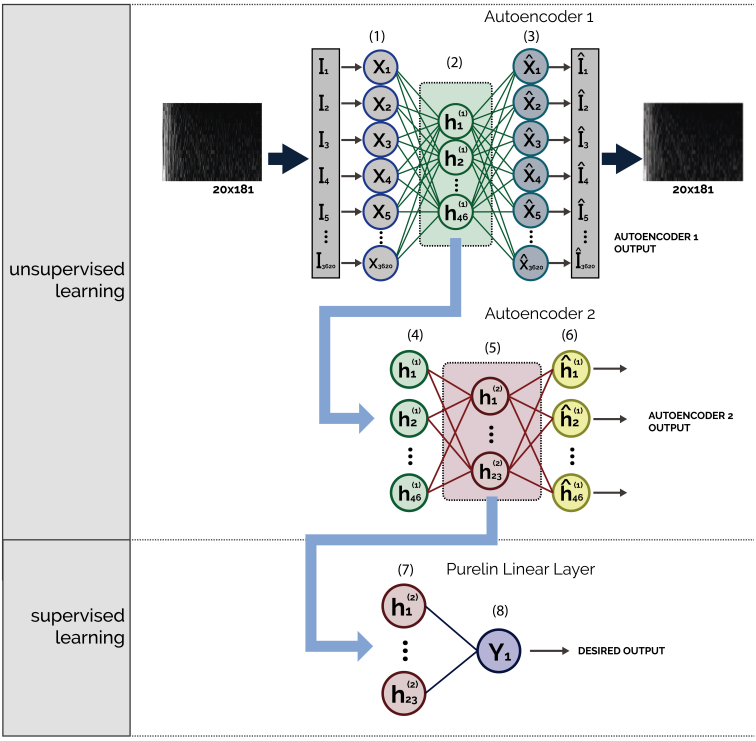


Figure 26: Diagram of the neural network model developed with two Autoencoders and one network with linear transfer function separately. In the unsupervised stage, we have two Autoencoders networks. In the supervised stage, we have the network with linear transfer function called Purelin. The blue arrows show the three steps connected through the hidden layer of the predecessor through the input layer and, in particular, Autoencoder 1 initiates the process with an input layer from the samples generated by the preprocessing steps (see Step 1 to 9 of Figure 25). Initially, we construct and train separately to reconstruct each step in a final network with two intermediate layers and an output layer, as shown in Figure 27.

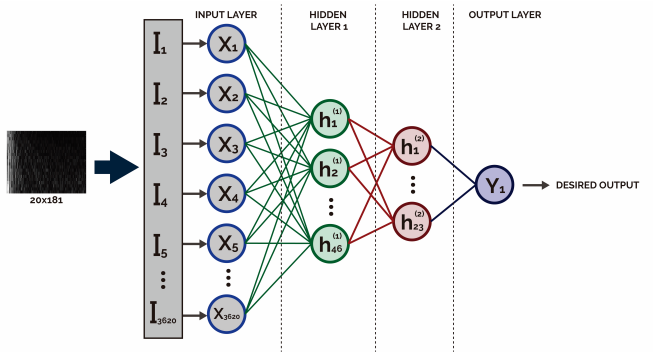


Figure 27: Network model constructed from the structures of the two Autoencoders and the network with linear transfer function (see Figure 26). In this way, we have a network with more than one hidden layer and a output layer already pre-trained, due to the adjusted weights and already allocated to each layer. Thus, a final training with the complete structure is done in order to refine the training before the final test with samples from the test dataset.

In Figure 26, we can see the structure of each Autoencoder and the network with linear transfer function in output layer. The first unsupervised stage consists of two Sparse Autoencoders networks, and the second stage consists the network with the linear transfer function. The three networks were built and trained separately. The first Autoencoder was trained with the principal dataset input reserved for the training stage with the 2DLS pattern particle images which received the preprocessing steps as shown in steps 1 to 8 in Figure 26. The number of neurons in the input of the first Autoencoder is 3620 because it is the size of the input vector of each sample of the training dataset, ie a 20x181 matrix transformed into a 1x3620. The hidden layer consists of 46 neurons of Autoencoder 1 and 23 in the hidden layer of Autoencoder 2 as shown in Table 3 with other parameteres. The choice of the number of neurons was given by the performance from several tests with different configurations.

In Figure 27 we see the final network with the desired output constructed from the structure of the other three previous networks (functions, connections and number of neurons) as well as the biases and pre-adjusted weights performed by each network in sequence.

In Table 3 we see the configuration of the two Autoencoders. The activation function used in both Autoencoders is the sigmoid function.

The Lambda (λ) coefficient is used in the Weight Regularizer term and the Beta (β) coefficient is used in the Sparsity Regularization term, as shown in the equation 3.3.

Table 3: Parameter values used in the two Autoencoders

Autoencoder	Number of neurons	Activation function	Lambda (λ)	Beta (β)
First one	46	Sigmoid	0.004	2
Second one	23	Sigmoid	0.002	2

Thus, we have our model using concepts of Deep Learning by a technique called DBN, which performs pre-training by separate networks that will later be the intermediate layers and the output layer of a Deep Learning network in order to extract characteristics not obtained by the previous models in order to have a better performance and in a way that is computationally feasible (COATES; LEE; NG, 2011; ZHANG; LI; ZHU, 2015).

This model achieved more satisfactory results than the other proposed models, mainly for the prediction of smaller particles, between 0 and 20 μm, ie as shown in Figure 28 with $R^2 = 0.99$.

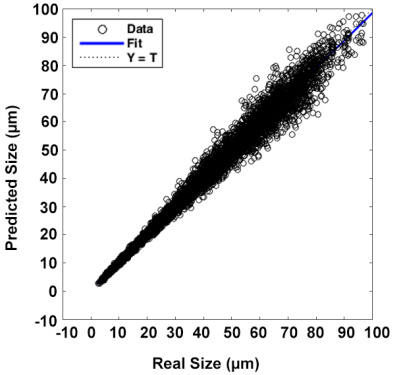


Figure 28: Plot showing predicted size with projected size (real size) of test process with model 4. Each data point represents a pattern image in a particle orientation. The x-axis shows the projected size of each orientation of the particles and the y-axis shows the size predicted by the the model. We see that the $R^2 = 0.991$ and that the smaller particles (between 0 and 20 μm) get an excellent prediction.

4 EXPERIMENTS AND RESULTS

In order to validate and compare the four models developed to verify which one could return the best prediction of a projected particle size, we executed the experiments using the same dataset of the 21,546 patterns, that is, samples of the particles used in this current work, which 70% (15,082 samples) are used for training and 30% (6,484 samples) for testing (see sections 3.1 and 3.3). The performance measure used to compare the models is the coefficient of determination (R^2) and the tests were executed in $MATLAB^{®}$ software.

In some experiments we used some specific particles in order to better understand the generalization problems of some of the models and thus to decide which new model will be develop or follow path to solve the main problem of prediction of smaller particle size, ie the particle with 5.3μm of length which is small and square particle. It is called particle 112. This particle was chosen also because it showed one of the worse results in the plot regression of the size prediction when all the particles were trained and tested.

The model 1 obtained satisfactory performance results around 0.98. However, after running this model several times, we found that network had difficulties to predict the size of small particle (ie, which have a small real size between 0 and 20 μm), as shown in Figure 27. We believe that one of the main reasons for this failure is that these particles have small variation in size and in diffraction (intensity) along of their 133 orientations, which could make the generalization of network more inefficient. So to check this, we have run new experiments trying to identify the behaviour of the particles which have small variation in size and intensity, and consequently were difficult to be training and tested by the network.

As shown in Figure 29 and Figure 30, when a small particle with little variation of diffraction was used in the testing set but it was not trained previously, it was more difficult for the network to predict its size.

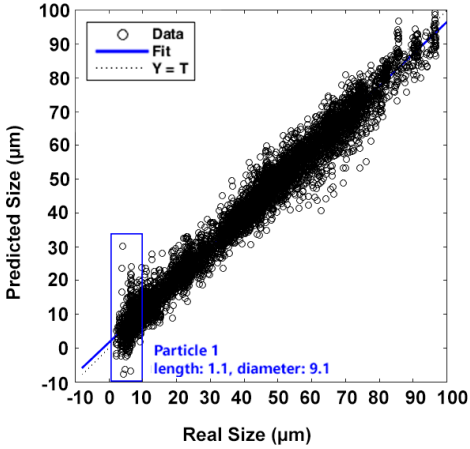


Figure 29: Plot of prediction results for the test set with model 1. The blue rectangle shows the particle which were difficult to be predicted by the network.

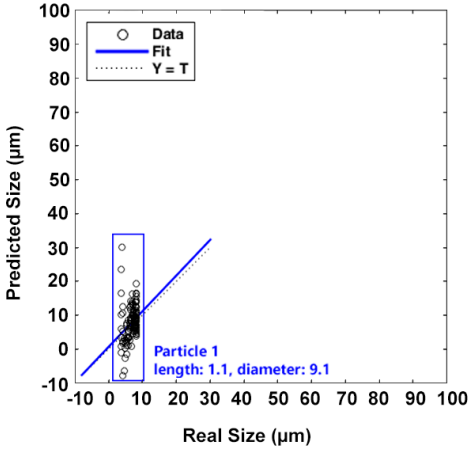
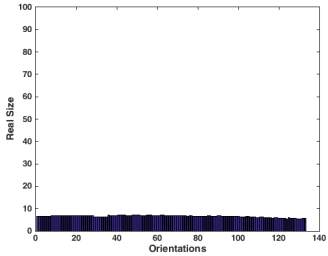
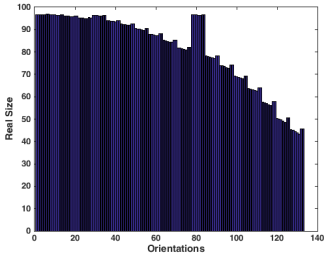


Figure 30: Prediction result for particle 1 used in the test set. This particle has 1.1 μm of length and 9.1 μm of diameter. That is a relatively small particle with little variation in size along its orientations, as it is shown in the x-axis.

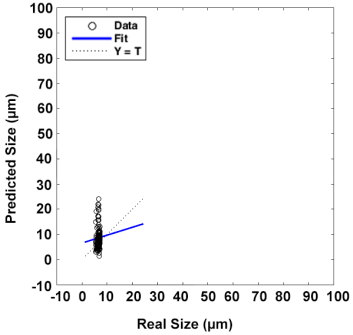
We realised that for the small particles which are also square (or compact) particle (i.e., they have a small difference between their length and diameter), when the size difference between its orientations is small, it is more difficult for the network to generalize and predict this type of particle when compared with other particles. To illustrate this, the Figure 31 shown comparisons of a smaller particle and a larger particle. Each graph shown all the 133 orientations of a unique particle (one of 162 particles of all dataset).



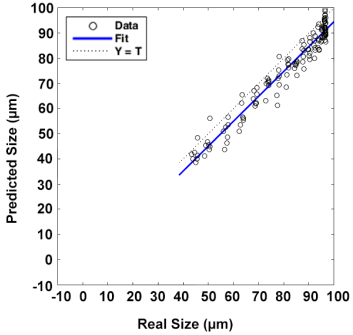
(A) Real sizes through its 133 orientations of a small particle. Y-axis is the real size (in μm) and the x-axis is the number of orientations



(B) Real sizes through its 133 orientations of a large particle. Y-axis is the real size (in μm) and the x-axis is the number of orientations



(C) Particle with small range in real size



(D) Particle with large range in real size

Figure 31: Results from a small particle (A and C graphs) and a large particle (B and D graphs).

As shown in Figure 31, we can see that the difference in size between the orientations of the small particle (A and C) is much smaller when compared to the difference in size of the large particle orientations (B and D). In some cases, the pattern of some samples of these smaller particles are very similar to particles of very different sizes (as we can see in Figure 15 of section 3.6.1) which there is a difficulty in generalization.

To analyse the results discussed above we have conducted further experiments, trying to identifying other particle features that could be related to the success and failure of the network prediction and are described in the next sections.

4.1 ENHANCEMENT OF CONTRAST INTENSITIES

Intensity variations of the particle patterns is relatively small to most of the small particles and particularly to those which tend to be square/compact compared to large particles. Thus, we conducted an experiment to check if increasing the contrast of the particle diffraction images could improve the network training and testing performances (see Figure 32).

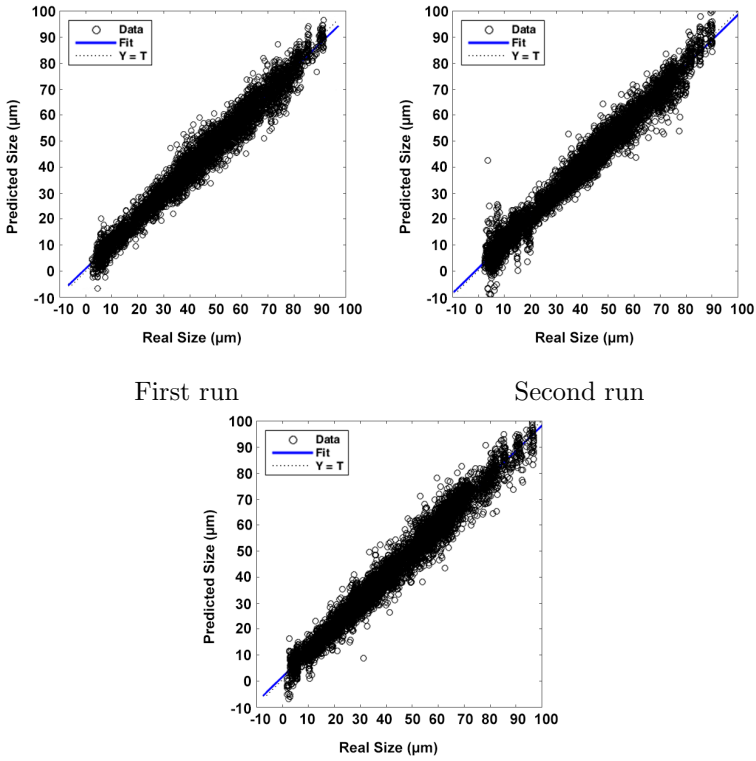


Figure 32: Runs of the network with diffraction contrast enhancement of particles images. Its general performances are $R^2 = 0.98$. However, there are several outliers.

In this experiment we maintained model 1 as the structure of the tests. We only added one more step between step 1 and step 2 that we visualize in Figure 12 of model 1. This further step is a contrast enhancement in each training sample and test by 20%. However, enhancing the contrast of diffraction images did not improve the network training and testing performance with this model and created several outliers as well.

4.2 TRAINING AND TEST PARTICLES INDIVIDUALLY

As the network was not able to generalize and predict some particles with specific characteristics (which was unclear to us), we have decided to start another strategy, by only training and testing the particles that apparently were always difficult to have their size predicted by the network. So, the purpose of this experiment was to find if the network have any problem in predicting this type of particle alone. Thus, we chose the particle named 112, which is a square and small particle with length and diameter equal to 5.3 μm (see Figure 33). In Figure 34, we show the results of the network when this problematic particle was trained and tested alone.

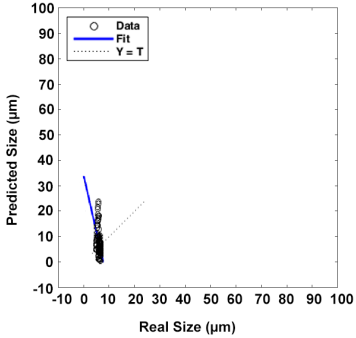


Figure 33: Network results (with $R^2 = 0.42$) for a square small particle (particle 112) when the network was trained with all training dataset and tested with all testing dataset.

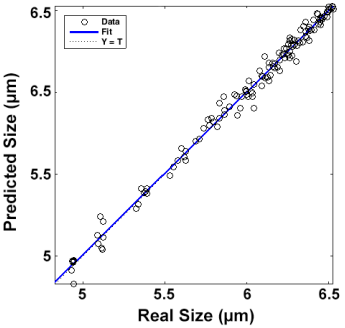


Figure 34: Plot test results for the particle 112 with all its 133 orientations (axes scale between 4.6 and 6.5 μm) with $R^2 = 0.995$.

When we use particles with distinct features to train and test the same network, this network usually has problems in predicting these type of particles; however, when we isolated these particles and test then in a different network for each particle, the network performs much better than before.

Thus, from this experiment, we realised that what was needed next was to identify which particles were considered "problematic", trying to identify the pattern of diffraction image that they can generate. As a result, we could isolate these particles, and train and test them separately from the other particles, which can mean using a single network to train and test the particle in a different way. From these results, we expect to find that not only the small square particles, with little variation of diffraction images, are the ones that influence the capacity of network in generalization, as other features and parameters can also affect the network performance. Thus, we decide to develop the model 2 with two NNs for prediction and one for classification to verify if separating those types of particle will increase the performance of the prediction.

4.3 EXPERIMENTS WITH MODEL 2

We consider that the particle samples that are problematic got error above 20% (Mean Squared Error) in the prediction with model 1 proposed in this current work. Because in most of the tests and experiments that we performed, samples with a prediction error above 20% we can already disregard as acceptable predictions for the proposed objectives.

Thus, model 2 uses one NN to classify samples between "non-problematic"and "problematic"(with Mean Square Error above 20%) and two NNs for prediction which are trained with already predetermined patterns as "non-problematic"and "problematic". NN 1 then trains and tests with patterns considered "non-problematic"and the NN 2 with the "problematics". Thus, the second network trains with fewer patterns, which are mostly smaller particles and could possibly have samples with similar patterns and can achieve a better generalization, in theory.

However, now the difficulty begins with the classification that becomes incorrect in most of the tests performed (see Figure 35), where the two NNs receive patterns that are quite different from those that were trained, as shown in Figure 36, making it difficult for prediction and generalization.

Figure 35: Confusion matrix of model 2 supervised NN for classification between problematic and non-problematic patterns. In this confusion matrix we see two lines. In the first line, is the classification of the orientations for NN 1 (the network with patterns not considered problematic). In the second line, samples that are possibly to NN 2 (the network trained to predict patterns considered problematic). It should be noted that some samples were classified into the wrong network (ie, 1330 samples should be in NN 2, but they were placed in NN 1). The total performance was 77.6%. That is, many samples were placed in networks that were not trained with similar patterns that would aid in generalization.

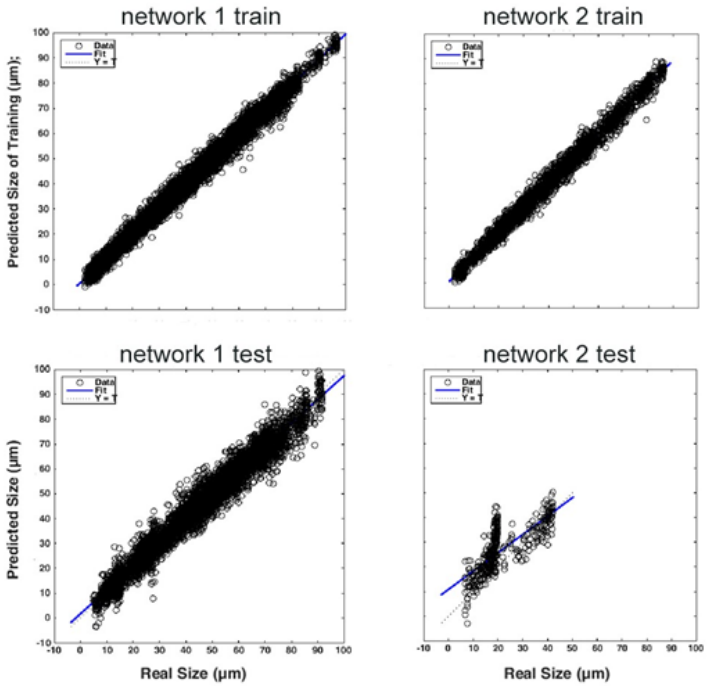


Figure 36: Training and test plots of the two prediction networks of model 2. In the upper left and right, we see the prediction in the training of NN 1 and 2 performed well. But, the test of both shown in the bottom left and right did not perform well. In the test of NN 1, we have a $R^2 = 0.97$. However, in NN 2, we have a R^2 around 0.80. In the y-axis, we see a prediction size (in μm). On the x-axis we see the projected real size (in μm).

4.4 EXPERIMENTS WITH MODEL 3

Model 3, as demonstrated in section 3.6.3 which details the proposed model, contains the preprocessing steps identical to model 1, with normalizations, Fourier transform and extraction of the first 40 components by PCA. The difference is in the NN used for training and testing, which we chose to create a network model by RBF, in order to be able to control in a more detailed way the structure, activation of the neurons and their organization. However, the predictions for the

smaller particles only obtained better results with the quantity around 5,000 neurons and with spread of each neuron having the coefficient of activation between 0.013 and 0.03, which is a very narrow spread, causing each neuron to have a greater ability to specialize in a more specific set of patterns presented for training, as we can see in Figure 37 with the test samples. However, by increasing the number of neurons even more, it often occurs overfittings, that is, the network tends to adapt to specific and unimportant details of the training set and causing poor generalization, which may reduce the prediction correction rate (ER et al., 2002).

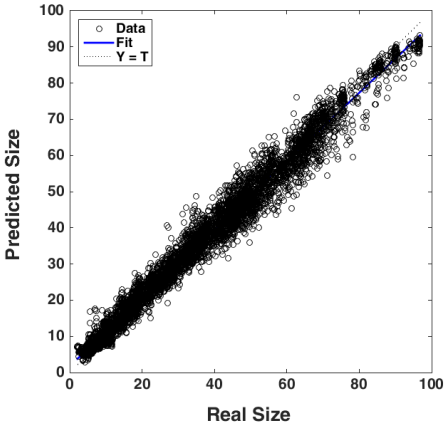


Figure 37: Test results using RBF neural networks with 5,000 neurons and small spread coefficient of each neuron (between 0.013 and 0.03). The $R^2 = 0.9902$.

However, the results of some small particles are still not satisfactory, that is, there is a significant difference between the bounds of predicted sizes and the real sizes of these particles, as we can see in Figure 38 with the isolated prediction (extracted from Figure 37) of the small square particle with 5.3μm of length. In other words, the R^2 of this particle is 0.1409.

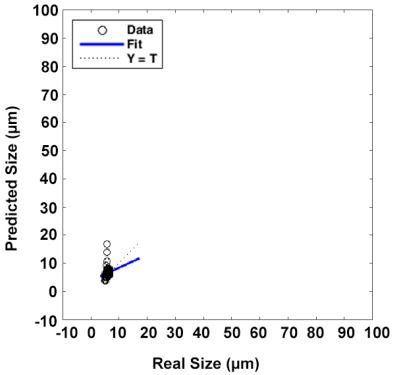


Figure 38: Test results of a squared small particle (5.3μm) using RBF neural networks with 5,000 neurons and small spread coefficient of each neuron (between 0.013 and 0.03). The R^2 of this particle is 0.1409.

4.5 EXPERIMENTS WITH MODEL 4

In model 4 we used pre-training steps with two unsupervised Sparse Autoencoders NNs and a supervised NN with linear transfer function; next we made a new train with a feed-foward network which is built with the structures of the three previous networks (hidden layers, adjusted weights, bias and functions). We achieve the most satisfactory results, especially in relation to the prediction of the smaller particles, as we can see in Figure 39.

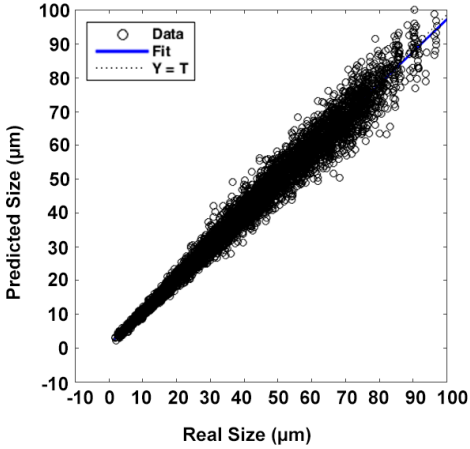


Figure 39: Test results with model 4. The $R^2 = 0.9914$ and the smaller particles was a satisfactory prediction (see Figure 40).

In Figure 40 we can verify the isolated prediction (extracted from the prediction shown in Figure 39) of small and square particle with 5.3μm of length.

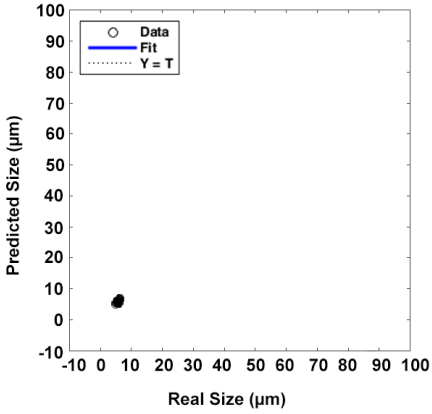


Figure 40: Test results with model 4 of the smaller and squared particle with 5.3μm of length.

In Figure 40 the $R^2 = 0.677$, but visually is good and satisfactory, because we can see little variation of prediction of the size, especially

if compare in the other models to the same particle (see section 4.6 for comparisons and discussion).

4.6 COMPARISONS AND DISCUSSION

From the results of the experiments, some observations can be made about the four proposed models. Table 4 shown the performance values of each particle in test dataset. Table 5 shown the performance values (Determination's Coefficient) of each model, as well as the error index Normalised Mean Squared Error (NMSE). In Figure 41 we have a comparison of the prediction of the four models with all test dataset. In Figure 42 we see a comparison of prediction of a small and square particle isolated from the others, with 5.3μm of length and diameter. In Figure 43 we have a comparison of prediction of a larger and not square particle.

Table 4: Performance values of each particle in test dataset for all models. It is in ascending order by the mean projected particle size from their length and diameters. (*P for Particle; L for Length; D for Diameter*)

Particles	Performance (R^2)			
	Model 1	Model 2	Model 3	Model 4
P 2 - L: 1.5 — D: 8.5	0.638	0.783	0.623	**0.959**
P 1 - L: 1.1 — D: 9.1	0.747	0.849	0.423	**0.956**
P 112 - L: 5.3 — D: 5.3	0.651	0.671	0.140	**0.881**
P 26 - L: 13.0 — D: 2.4	0.525	0.674	0.477	**0.976**
P 99 - L: 4.1 — D: 12.3	0.06	0.172	0.372	**0.932**
P 114 - L: 5.6 — D: 11.1	0.371	0.170	0.284	**0.882**
P 37 - L: 15.7 — D: 2.0	0.346	0.456	0.734	**0.974**
P 147 - L: 8.9 — D: 8.9	0.373	0.295	0.439	**0.776**
P 39 - L: 15.8 — D: 5.3	0.340	0.342	0.728	**0.969**
P 115 - L: 5.6 — D: 16.9	0.530	0.589	0.793	**0.926**
P 83 - L: 3.5 — D: 19.2	0.786	0.825	0.829	**0.974**
P 75 - L: 26.9 — D: 3.4	0.840	0.299	0.899	**0.989**
P 38 - L: 15.8 — D: 15.8	0.371	0.348	0.547	**0.896**
P 85 - L: 3.9 — D: 31.6	0.886	0.882	0.952	**0.987**
P 146 - L: 8.9 — D: 26.6	0.711	0.773	0.749	**0.964**

Continued on next page

Tabela 4 – continued from previous page

Particles	Performance (R^2)			
	Model 1	Model 2	Model 3	Model 4
P 87 - L: 31.6 — D: 5.7	0.797	0.681	0.900	**0.981**
P 79 - L: 29.1 — D: 9.7	0.739	0.582	0.778	**0.970**
P 101 - L: 4.6 — D: 37.2	0.933	0.862	0.880	**0.981**
P 31 - L: 14.0 — D: 28.1	0.656	0.536	0.449	**0.805**
P 77 - L: 28.3 — D: 14.2	0.828	0.565	0.888	**0.957**
P 134 - L: 7.4 — D: 40.5	0.905	0.853	0.934	**0.980**
P 124 - L: 6.7 — D: 53.9	0.949	0.936	**0.972**	0.969
P 44 - L: 16.6 — D: 49.9	0.833	0.849	**0.893**	0.884
P 66 - L: 22.6 — D: 45.2	0.731	0.687	0.738	**0.776**
P 121 - L: 59.3 — D: 10.8	0.757	0.825	0.892	**0.986**
P 71 - L: 24.7 — D: 49.4	**0.760**	0.679	0.689	0.746
P 119 - L: 55.8 — D: 18.6	0.896	0.803	0.872	**0.954**
P 98 - L: 39.7 — D: 39.7	0.619	0.614	0.621	**0.761**
P 131 - L: 68.5 — D: 12.5	0.848	0.813	0.909	**0.964**
P 118 - L: 54.6 — D: 27.3	0.882	0.753	**0.889**	0.874
P 104 - L: 43.2 — D: 43.2	0.716	0.695	0.687	**0.830**
P 68 - L: 23.0 — D: 69.0	0.907	0.852	**0.939**	0.916
P 5 - L: 10.3 — D: 82.6	0.969	0.942	**0.988**	0.987
P 106 - L: 46.5 — D: 46.5	0.763	0.777	**0.779**	0.749
P 150 - L: 83.6 — D: 10.4	0.883	0.892	0.965	**0.968**
P 70 - L: 24.5 — D: 73.5	0.900	0.891	**0.950**	0.927
P 13 - L: 11.7 — D: 93.6	0.980	0.973	**0.989**	0.981
P 159 - L: 94.9 — D: 11.9	0.936	0.913	0.977	**0.982**
P 20 - L: 12.3 — D: 98.5	0.970	0.974	**0.987**	0.964
P 27 - L: 13.2 — D: 105.5	0.960	0.946	0.985	**0.987**
P 148 - L: 81.1 — D: 40.5	**0.924**	0.886	0.907	0.877
P 8 - L: 105.7 — D: 19.2	0.935	0.911	0.945	**0.958**
P 160 - L: 95.6 — D: 31.9	0.911	0.918	**0.935**	0.923
P 18 - L: 118.0 — D: 14.7	0.948	0.950	**0.959**	**0.959**
P 16 - L: 115.3 — D: 21.0	0.941	0.931	0.949	**0.957**
P 7 - L: 102.5 — D: 34.2	0.927	0.894	**0.942**	**0.942**
P 10 - L: 108.9 — D: 36.3	**0.945**	0.935	**0.945**	0.944
P 17 - L: 115.6 — D: 38.5	0.940	0.915	**0.944**	0.887
P 34 - L: 143.2 — D: 26.0	0.960	0.963	**0.974**	0.944
Performance mean	0.7688	0.7413	0.7967	**0.9247**

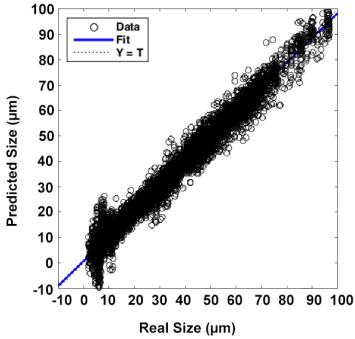
In Table 4 we show that almost all the particles presented better prediction with model 4, especially the smaller ones. But in some cases, larger particles had a better result in model 3, or 1. In general, the best results were found between models 3 and 4 when predicting the size of larger particles. The hypothesis of this behavior (best results with model 4) is discussed in more details in chapter 5.

However, some particles had not a good performance in prediction (up to 0.8) in all models, such as Particle 66, 98 and 147. Other particles had the worst prediction between model 1 and 3 and a good performance in model 4, such as Particle 99.

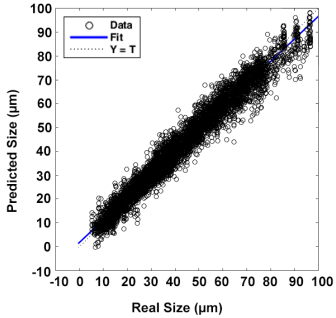
The hypothesis of this behavior in this cases is the aspect ratio and their pattern images which could influence the performance, which in some cases can be very similar to another pattern image from a different particle with a big difference in size value (as shown in Figure 15 at section 3.6.1). This behavior can be another object of study for future works.

Table 5: Performance and error values of each model

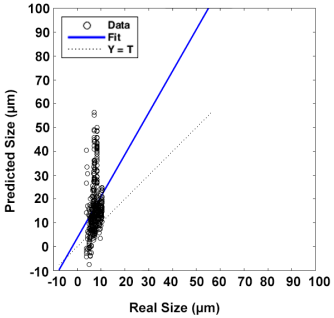
MODEL	Performance (R^2)	NMSE
MODEL 1	0.9853	0.0059
MODEL 2	NN 1: 0.9812 — NN 2: 0.2451	NN 1: 0.0075 — NN 2: 3.61
MODEL 3	0.9891	0.004
MODEL 4	**0.9914**	**0.003**



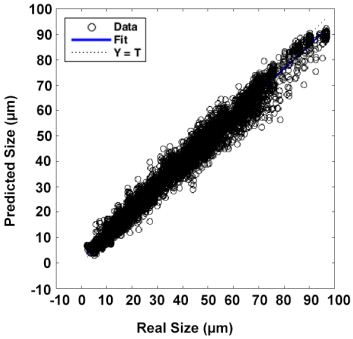
Model 1



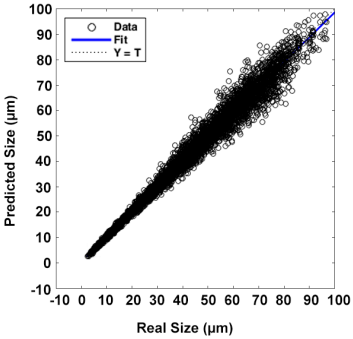
Model 2 NN 1



Model 2 NN 2

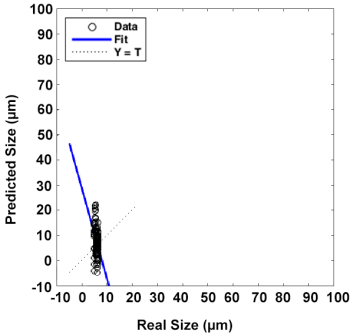


Model 3

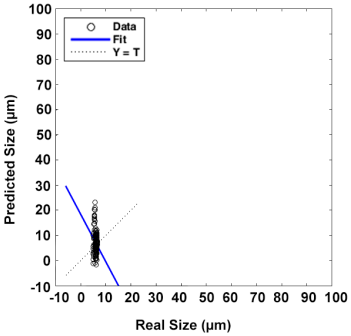


Model 4

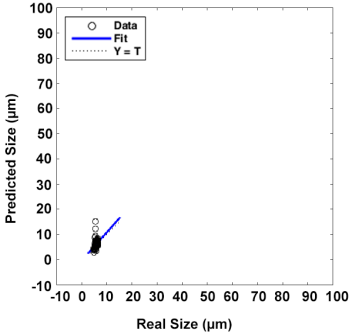
Figure 41: Prediction comparison of all models with all test dataset.



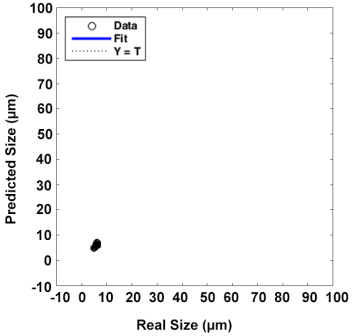
Model 1 - $R^2 = 0.31$



Model 2 - $R^2 = 0.18$

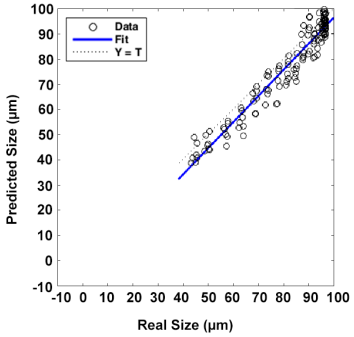


Model 3 - $R^2 = 0.34$

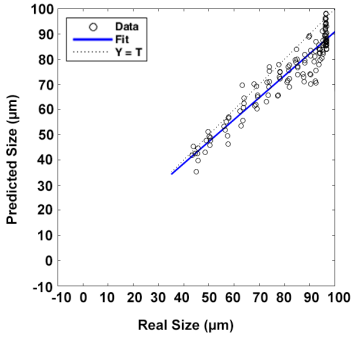


Model 4 - $R^2 = 0.81$

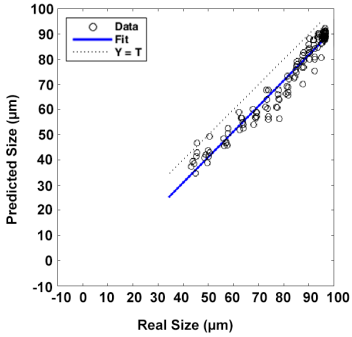
Figure 42: Prediction comparison of all models with a smaller squared particle (5.3μm of length).



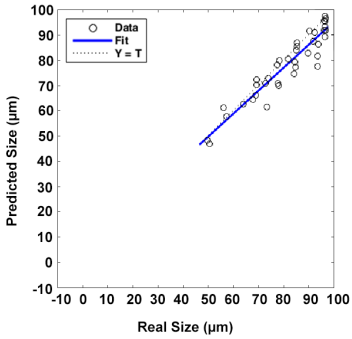
Model 1 - $R^2 = 0.968$



Model 2 - $R^2 = 0.946$



Model 3 - $R^2 = 0.976$



Model 4 - $R^2 = 0.98$

Figure 43: Prediction comparison of all models with a larger particle (length: 13.2μm; diameter: 105.5μm.)

As can be seen in Table 5, Figure 41, Figure 42 and Figure 43, that model 1 obtained a prediction result, in values of percentage, considered acceptable (R^2 = 0.9853 and NMSE = 0.0059). However, as discussed in this work, the prediction for small particles would need to be better and therefore the development of other techniques and data processing would be necessary.

Model 2 had the worst performance, with R^2 = 0.9812 (NMSE: 0.0075) in the NN 1, which trains and tests patterns considered "non-problematic", and R^2 = 0.2451 (NMSE: 3.61) for the NN 2, which trains and tests patterns considered "problematic".

Model 3 with RBF obtained relatively better results compared to model 1 and 2 (R^2 = 0.9891 and NMSE = 0.004). However, because of the amount of neurons needed to perform better on smaller particles it may have generated consequences as overfitting in several tests. Therefore, it was difficult to find a spreads and neurons configuration that would contribute to a good prediction performance and avoid an overfitting.

Model 4 obtained the best performance in the prediction mainly with the smaller particles. As shown in Table 5, the highest performance index achieved was using model 4 (R^2 = 0.9914). The NMSE in model 4 is much less than others (0.003).

In Table 6 we show other statistic variables in order to understand in more details the performance of this prediction.

Table 6: Statistic of the model 4 prediction

Performance (R^2)	Confidence Interval	Confidence Level
0.9914	0.99098 - 0.99182	95%

As shown in Table 6, the confidence interval is between 0.99098 and 0.99182 with 95% of confidence level. That is, this model has a prediction with good confidence interval and it is strongly correlated with the real size values.

5 CONCLUSIONS

In this work, four models with neural network techniques were proposed in order to predict the size of atmospheric particles captured by 2DLS patterns. A comparison was made between the proposed models and raised the possible hypotheses that led to the development of more than one model to solve the problem of smaller particles prediction.

The results showed that some projected particle sizes, especially smaller ones, are more difficult to predict when compared to larger ones. These results were also found by other authors with spherical particles, as discussed in Ulanowski et al. (1998). One possible reason, as discussed in section 3.6.1, is that some particles with considerably different sizes may have very similar diffraction patterns (see Figure 14). It has been found that most problematic orientations appear in small particles, which may result from the amount of incident light scattered by small particles being smaller than for larger ones and the scattered light is wider.

5.1 MAIN CONTRIBUTIONS

It is possible to note that the model 2 which uses MLP feed-forward network for classification has significant difficulties to classify the patterns for the two regression NNs, as well as the prediction of NN 2 that even receiving some pattern correctly, the test was not effective. A hypothesis for this result again declines to the fact that some patterns are similar, but are from very different particle with different sizes, making it difficult to generalization in the models developed here.

We believe that model 3 was better performance than model 1 because the Gaussian functions, when we configure correctly, may be able to specialize better in every small difference that occurs in patterns that are extremely divergent in sizes. In other words, using RBF with more controls inside the neurons and its spreads and activation, even with a large number of neurons, maybe we can control more the generalization and avoid overfitting, if made correctly.

However, using Deep Learning techniques with Sparse Autoencoders NNs we believe also that are better methods to extract the characteristics of the particles patterns than PCA. Thus, the PCA process may not maintain all the fundamental features of these patterns

from 2DLS, even increasing considerably the amount of the principal components used for training and testing process of the neural network models presented in this work. This behavior was discussed in Hinton e Salakhutdinov (2006), which demonstrated that Autoencoders can be more effective than PCA for dimensional reduction and feature extraction (HINTON; SALAKHUTDINOV, 2006).

In other words, comparing model 4, that use Autoencoders, with the results of the other three models that used PCA, we found that some characteristics of the particles captured through 2DLS patterns have only become evident through a distributed representation, that is, through learning between intermediate layers and resulting from more than one sample, especially with smaller particles. Thus, it was possible to detect higher-level abstractions that characterize the input and that perhaps could not be extracted from traditional models, such as a model using MLP feed-forward with a hidden layer. This behavior in the field of image pattern recognition was observed in the works of Seyyedsalehi e Seyyedsalehi (2015), Zhang, Li e Zhu (2015), Wang et al. (2015) and other works analyzed using concepts of Deep Learning and pre-training by Autoencoders.

Also, this techniques can be applied in a computationally viable time and to be able to have a better performance in the test phase. In Table 7 we show the computational time in training and testing process using a PC with a Intel® Core^{TM} i7 2.00GHz processor and 8GB RAM.

Table 7: Computational time using model 4 (in seconds)

Network	Training Phase	Testing phase
First Autoencoder	6012	0.71
Second Autoencoder	28	0.11
Linear Function Network	3	0.02
Final Network	3668	1.29
Total	9771	2.13

As shown in Table 7, the time of all training set (sum of both Autoencoders, Linear Function Network and Final Network training phases) was around two hours and forty minutes (9771 seconds); Final Network testing phase lasted around 1.29 second. Although the training phase takes a relatively long time, if made correctly, it needs to be done once with the trainig dataset. However, only the testing phase

is used in practical purposes.

Thus, the aim of this work, which is to propose a computational model using neural networks (NN) that can effectively predict the projected size of atmospheric particles, has been achieved mainly in smaller ones using model 4. However, in larger particles, both model 3 and 4 has similar and satisfactory results.

5.2 FUTURE WORKS

As a future work, we can further study the behavior of 2DLS particle patterns and analyse in details the techniques of Deep Learning with Autoencoders and Convolutional Networks in order to better predict also the larger particles which, although satisfactory, can have a better result.

Another interesting future work is to improve model 4 to extract other characteristics of the particles studied in this work, such as aspect ratio, angle of orientation and other pertinent characteristics for other fields of study, such as atmospheric science, climatology, among others.

REFERENCES

BARAN, A. J. A review of the light scattering properties of cirrus. **Journal of Quantitative Spectroscopy and Radiative Transfer**, v. 110, n. 14-16, p. 1239–1260, 2009. ISSN 0022-4073.

BEAUDOIN N.BEAUCHEMIN, S. An accurate discrete fourier transform for image processing. **Object recognition supported by user interaction for service robots**, 2002.

BENGIO, Y. Learning deep architectures for ai. **FNT in Machine Learning**, v. 2, n. 1, p. 1–127, 2009.

BHOWMIK, M. K. et al. Classification of fused images using radial basis function neural network for human face recognition. In: **2009 World Congress on Nature Biologically Inspired Computing (NaBIC)**. [S.l.: s.n.], 2009. p. 19–24.

CLIMATE Change 2013: The Physical Science Basis. 2013. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change.

COATES, A.; LEE, H.; NG, A. Y. An analysis of single-layer networks in unsupervised feature learning. In: **AISTATS 2011**. [S.l.: s.n.], 2011. v. 1001.

EL-BAKRY HAZEM M.MASTORAKIS, N. New fast normalized neural networks for pattern detection. **Image and Vision Computing**, v. 25, n. 11, p. 1767–1784, 2007.

ER, M. J. et al. Face recognition with radial basis function (rbf) neural networks. **IEEE Transactions on Neural Networks**, v. 13, n. 3, p. 697–710, May 2002. ISSN 1045-9227.

FORESEE, F. D.; HAGAN, M. T. Gauss-newton approximation to bayesian learning. In: **Neural Networks,1997., International Conference on**. [S.l.: s.n.], 1997. v. 3, p. 1930–1935.

GEORGE, M. Radial basis function neural networks and principal component analysis for pattern classification. In: **International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)**. [S.l.: s.n.], 2007. v. 1, p. 200–206.

GRIFFITHS, D. **Introduction to Electrodynamics**. [S.l.]: Prentice Hall, 1999. ISBN 9780138053260.

GUARDANI, R.; ONIMARU, R.; CRESPO, F. Neural network model for the on-line monitoring of a crystallization process. **Brazilian Journal of Chemical Engineering**, scielo, v. 18, p. 267 – 275, 09 2001. ISSN 0104-6632.

GUGLIOTTA, L. M. et al. A neural network model for estimating the particle size distribution of dilute latex from multiangle dynamic light scattering measurements. **Particle & Particle Systems Characterization**, WILEY-VCH Verlag, v. 26, n. 1-2, p. 41–52, 2009. ISSN 1521-4117.

HESSE, E. et al. Application of rtdf to particles with curved surfaces. **Journal of Quantitative Spectroscopy and Radiative Transfer**, v. 110, n. 14-16, p. 1599–1603, 2009. ISSN 0022-4073.

HINTON, G. E.; SALAKHUTDINOV, R. R. Reducing the dimensionality of data with neural networks. **Science**, American Association for the Advancement of Science, v. 313, n. 5786, p. 504–507, 2006. ISSN 0036-8075. Disponível em: <http://science.sciencemag.org/content/313/5786/504>.

HODGSON, R. Genetic algorithm approach to particle identification by light scattering. **Journal of Colloid and Interface Science**, v. 229, n. 2, p. 399 – 406, 2000. ISSN 0021-9797. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0021979700969893>.

KAYE, P.; HIRST, E.; WANG-THOMAS, Z. Neural-network-based spatial light-scattering instrument for hazardous airborne fiber detection. **Appl. Opt.**, v. 36, n. 24, p. 6149, 1997.

KAYE, P. H. et al. Classifying atmospheric ice crystals by spatial light scattering. **Opt. Lett.**, OSA, v. 33, n. 13, p. 1545–1547, 2008.

LAWSON, R. et al. Improved measurements of the drop size distribution of a freezing drizzle event. **Atmospheric Research**, v. 47-48, p. 181–191, 1998. ISSN 0169-8095.

MACKAY, D. J. Bayesian interpolation. **Neural Computation**, v. 4, p. 415–447, 1991.

MACKAY, D. J. C. **Information Theory, Inference & Learning Algorithms**. New York, NY, USA: Cambridge University Press, 2002. ISBN 0521642981.

MIE, G. Beiträge zur optik trüber medien, speziell kolloidaler metallösungen. **Annalen der Physik**, WILEY-VCH Verlag, v. 330, n. 3, p. 377–445, 1908. ISSN 1521-3889.

OLSHAUSEN, B. A.; FIELD, D. J. Sparse coding with an overcomplete basis set: A strategy employed by v1? **Vision Research**, v. 37, n. 23, p. 3311 – 3325, 1997. ISSN 0042-6989. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0042698997001697>.

RIEFLER, N.; WRIEDT, T. Intercomparison of inversion algorithms for particle-sizing using mie scattering. **Particle & Particle Systems Characterization**, WILEY-VCH Verlag, v. 25, n. 3, p. 216–230, 2008. ISSN 1521-4117.

SALAWU, E. O. **Development of Computational Models for Characterizing Small Particles Based on their Two-Dimensional Light Scattering Patterns**. Dissertação (Mestrado) — University of Hertfordshire, Hatfield, UK, 2015.

SEYYEDSALEHI, S. Z.; SEYYEDSALEHI, S. A. A fast and efficient pre-training method based on layer-by-layer maximum discrimination for deep neural networks. **Neurocomput.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 168, n. C, p. 669–680, nov. 2015. ISSN 0925-2312. Disponível em: <http://dx.doi.org/10.1016/j.neucom.2015.05.057>.

SONG, F.; GUO, Z.; MEI, D. Feature selection using principal component analysis. **2010 International Conference on System Science, Engineering Design and Manufacturing Informatization**, 2010.

STOPFORD, C. **Ice Crystal Classification Using Two Dimensional Light Scattering Patterns**. Tese (Doutorado) — University of Hertfordshire, Hatfield, UK, 2010.

ULANOWSKI, Z. et al. Application of neural networks to the inverse light scattering problem for spheres. **Appl. Opt.**, OSA, v. 37, n. 18, p. 4027–4033, 1998.

WANG, Y. et al. Dimensionality reduction strategy based on auto-encoder. In: **Proceedings of the 7th International Conference on Internet Multimedia Computing and Service**. New York, NY, USA: ACM, 2015. (ICIMCS '15), p. 63:1–63:4. ISBN 978-1-4503-3528-7. Disponível em: <http://doi.acm.org/10.1145/2808492.2808555>.

ZHANG, Z.; LI, J.; ZHU, R. Deep neural network for face recognition based on sparse autoencoder. In: **2015 8th International Congress on Image and Signal Processing (CISP)**. [S.l.: s.n.], 2015. p. 594–598.